TWO ALGORITHMS FOR FINDING THE ABSOLUTE M-CENTER OF A GRAPH

by

John Jesse Reed

United States
Naval Postgraduate School

THESIS

TWO ALGORITHMS FOR FINDING
THE
ABSOLUTE M-CENTER OF A GRAPH

by

John Jesse Reed

Thesis Advisor: A. W. McMasters

March 1971

Two Algorithms for Finding the Absolute M-Center of a Graph

by

John Jesse Reed
Lieutenant, United States Navy
B.A., University of Utah, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1971

ABSTRACT

Two algorithms for finding the absolute m-center are developed, combining the ideas of Hakimi, Gillespie, and Rosenthal and Smith. The first algorithm developed is essentially a hand-computational method. It is based on partitioning the graph into m subgraphs centered on the elements of the vertex m-center. The minimum distance tree rooted on each element of the vertex m-center is then formed and modified to yield the central path and thus the absolute center of each subgraph. This algorithm will give the absolute m-centers of a graph if each of these m central paths passes through an element of the vertex m-center. The second algorithm is an iterative search of all possible sets of m edges on which the absolute m-center may be located. It is less efficient than the algorithm of Rosenthal and Smith when m = 1, but appears to be more efficient for m > 1. It does eliminate the problems encountered by the Rosenthal-Smith algorithm in handling peripheral vertices.

TABLE OF CONTENTS

I. INTRODUCTION

The optimum location problem has been examined in many contexts by various authors dating back as early as the 17th Century. (See Francis [2] for an extensive bibliography on this subject.) It arises, for example, in determining the best location for one or more communication, distribution service or emergency service facilities within a given area to be serviced. The "costs" to be minimized may be actual road construction costs as a function of distance, travel time as a function of distance, the actual distance itself, etc.

The location problem is modeled by a graph $G(V,A)$: where V is the set of vertices which correspond to the localities to be served, and A is the set of edges which correspond to the transportation or communication links interconnecting these locations. Fixed costs or distances are assigned to each edge of the graph. (See p. 27-30 of Ref. [6] for basic graph theory.)

The m-center problem has arisen in two different forms from efforts to apply graph theoretic techniques to the solution of optimum location problems. The first and by far the most easily solved form is the vertex m-center problem, applicable in cases where the facilities are constrained to be located at a vertex of the representative graph. This problem was defined and solved for $m = 1$ by Hakimi [4], and was extended to cases with $m > 1$ by Gillespie [3].

The research reported herein is directed toward the solution of the second form of the problem, the absolute m-center. The absolute m-center is applicable where the facilities to be located are only constrained to lie at some point of the graph G. The object then is to minimize the maximum distance from any locality to be served to the nearest service facility.

This report is based on the previous works of Hakimi [4,5] and Gillespie [3]; and, although conceived independently, it parallels some of the work of Rosenthal and Smith [7].

Section II introduces the appropriate graph-theoretic concepts and summarizes the solution techniques developed by Hakimi, Gillespie, and Rosenthal and Smith. Section III presents the discussion and formulation of a heuristic algorithm for finding the absolute center of a graph, and formulates but does not apply its extension to the absolute m-center problem. Section IV is devoted to the development of a second, more analytical algorithm for the solution of the absolute center problem and its extension to the absolute m-center. A brief example of its application is given in an absolute 2-center problem. Section V summarizes the report and suggests some areas for further consideration.

II. PAST WORK

A. THE VERTEX AND ABSOLUTE CENTERS OF A GRAPH

Consider a connected graph $G(V,A)$ consisting of n vertices (nodes) and $M \leq n - 1$ undirected edges (arcs). This graph may be a model of a communication or transportation system in which traffic is allowed to flow simultaneously in either direction along each edge, or branch of the system. Let $l(i,j)$ be the distance along the single edge (i,j) connecting adjacent vertices i and j, and let $d(i,j)$ be the minimum distance on G between any two vertices i and j. Similarly, $d(x,y)$ is the length of the shortest path on G between any two points x and y on G. It will also be useful to extend this notion to include several intermediate vertices; i.e., $d(i,j,k,u,v) = d(i,j) + d(j,k) + d(k,u) + d(u,v)$. Some intermediate vertices may be omitted where the omission will not cause confusion.

The n x n matrix $D = ||d_{ij}||$ is defined as:

$$d_{ij} = \begin{cases} d(v_i,v_j); \ i, \ j = 1, \ \ldots \ ,n, \ j \neq i; \\ d(v_i,v_i) = 0; \ i = j, \ i = 1, \ \ldots \ ,n \ . \end{cases} \tag{1}$$

Define the radius associated with a vertex $k \varepsilon V$ to be

$$r(k) = \max_{v \varepsilon V} \ d(k,v) \ .$$

The vertex center is then defined to be that vertex in G with minimum radius. This radius will be called the vertex-center

radius of G, and is obtained from

$$r_c = \min_{1 \leq j \leq n} \max_{1 \leq i \leq n} d(v_i, v_j) = \min_{v \varepsilon V} r(v) \quad (2)$$

Similarly, the absolute center of G is defined as that point x^* on G such that

$$\min_{x \text{ on } G} \max_{1 \leq i \leq n} d(v_i, x) = \max_{1 \leq i \leq n} d(v_i, x^*) = r_a. \quad (3)$$

1. <u>The Hakimi Algorithm</u>

In a first approach toward finding the absolute center of G, Hakimi [4] used equation (3) as the rationale for solving M simpler 'min-max' problems. The result is essentially a hand method requiring the plotting of M sets of linear distance functions, one set for each edge in A, to find M 'local centers', $x_1, x_2, \ldots, x_M$. The value of x_j on edge (p,q) is found by plotting the function

$$d(v_i, x) = \min\{[x + d(p, v_i)]; [l(p,q) - x + d(q, v_i)]\}, \quad (3a)$$

$i = 1, \ldots, n$, and finding the $\max_i d(v_i, x)$. That x_j, $j = 1, \ldots, M$, which minimizes (3) is then chosen as the absolute center, and its radius is the absolute radius.

This method is very tedious and time consuming for graphs containing a large number of vertices.

2. <u>The Rosenthal-Smith Algorithm</u>

Rosenthal and Smith [6] took a very different, analytical approach. They began by presenting the following theorem, the proof of which is elementary and is omitted here.

Theorem 1: The absolute center of graph G lies on the midpoint of some path which connects two vertices (not necessarily adjacent) of G.

They define the 'central path' of G to be the path (v_x, v_y, v_z) which satisfies

$$d(v_x, v_y, v_z) = \min_j \left[\max_{i,k} d(v_i, v_j, v_k) \right] \tag{4}$$

for $i,j,k = 1, \ldots, n$; $i \neq k$; where i and k index the columns and j the rows of the distance matrix D.

The object is to find the longest path through each vertex of G, and then to find the vertex having the shortest such path. These paths must be non-recursive; that is, in traveling from vertex i through j to k, each edge in the path is traversed only once.

The procedure is as follows: For every row in D (fixed value of j) find the two largest numbers (greatest distances) and sum them. Call this sum $d^*(v_i, v_j, v_k)$. Now find the minimum of all $d^*(v_i, v_j, v_k)$ as j goes from one to n. It is then necessary to check that the route found from i through j to k is a path for which each edge is traversed only once (is non-recursive). If the path found has 'backpaths' (is recursive), the second longest route involving j is found and the path checking is repeated. This process is continued until the longest non-recursive path through vertex j has been found. If this requirement to check each path for backpaths could be eliminated, the Rosenthal-Smith algorithm would be much more efficient.

The authors conclude their algorithm with the following theorem which describes the final step of the algorithm.

<u>Theorem 2</u>: The absolute center x* of graph G is located at the midpoint of the central path of G.

Proof: Let the midpoint of the central path of G be the point x_0. To show that x_0 is the absolute center of G one need only show that the midpoint of any other path in G will yield a larger radius; because by Theorem 1 the absolute center lies at the midpoint of some path in G.

The following example, originally from Hakimi [4] and also used by Rosenthal and Smith, will illustrate the latter's algorithm for the absolute center of G.

Consider the graph in Figure 1 with the distance matrix

$$D = \begin{bmatrix} 0 & 10 & 24 & 20 & 34 \\ 10 & 0 & 14 & 12 & 24 \\ 24 & 14 & 0 & 12 & 10 \\ 20 & 12 & 12 & 0 & 20 \\ 34 & 24 & 10 & 20 & 0 \end{bmatrix} \quad \begin{array}{cc} \underline{T+} & \underline{T} \\ 58 & 54 \\ 38 & 36 \\ 38 & 36 \\ 40 & 40 \\ 58 & 54 \end{array} .$$

First, compute $\max_{i,k} d(v_i,v_j,v_k)$; $i,k = 1, \ldots ,n$; $i \neq k$; for each row (vertex). This is done by adding the two largest elements in each row: For row 1 this gives 24 + 34 = 48 58 and corresponds to the path (v_3,v_1,v_5). These sums are listed in the column headed T+, shown as the first column to the right of the D matrix. Notice in Figure 1 that the route (v_3,v_1,v_5) is not a non-recursive path, as the paths (v_3,v_1) and (v_1,v_5) have the edges (3,2) and (2,1) in common. The

next longest route involving v_1 is (v_4, v_1, v_5) with length 54 and it is a non-recursive path; list the value 54 in the column headed T. The values in the T column correspond to the longest non-recursive path for each j. It is evident from a comparison between elements of columns T+ and T that the only non-recursive path found in the first step (T+ column) is (v_1, v_4, v_5). Now find the minimum entry in the T column. There are two rows (vertices 2 and 3) with the minimum value, 36; these correspond to the paths (4,2,5) and (4,3,1) with midpoints x and y in Figure 1, respectively. Both have the same absolute radius; $r_a = r(x) = r(y) = 18$.

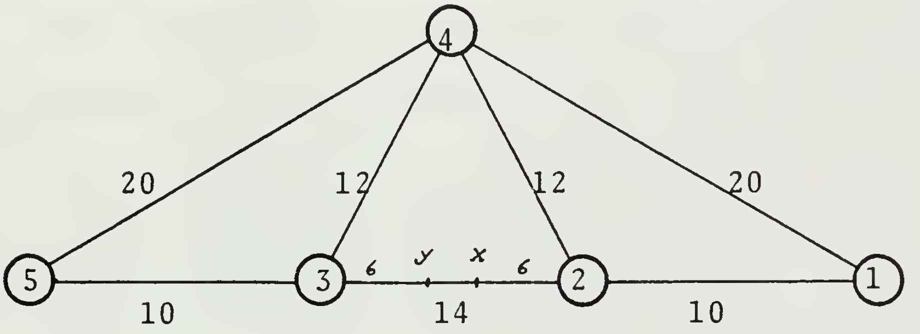


Figure 1: Example 1

B. THE VERTEX M-CENTERS AND ABSOLUTE M-CENTERS OF A GRAPH

1. The Vertex M-Center

Gillespie [3] extended the concept of the vertex center of graph G(V,A) to vertex multi-centers in the following manner. A set of m vertices $V_m^* \subseteq V$ is called a vertex m-center of G if, for every other set of m vertices $V_m \subseteq V$,

$$\max_{v\varepsilon V} d(v,V_m) \geq \max_{v\varepsilon V} d(v,V_m{*}) = r_m \;; \tag{5}$$

where r_m is defined as the radius associated with the vertex m-center.

Suppose a graph has the following distance matrix,

$$D = \begin{bmatrix} 0 & 2 & 5 & 3 & 5 & 4 \\ 2 & 0 & 3 & 4 & 5 & 6 \\ 5 & 3 & 0 & 3 & 2 & 4 \\ 3 & 4 & 3 & 0 & 5 & 7 \\ 5 & 5 & 2 & 5 & 0 & 2 \\ 4 & 6 & 4 & 7 & 2 & 0 \end{bmatrix} .$$

Since there are six vertices there are $\binom{6}{2} = 15$ sets of two vertices to be examined in finding the vertex 2-center; in general there are $\binom{n}{m}$ sets of m vertices to be examined for the m-center problem.

First, determine the minimum distance from each pair of vertices to every other vertex. Thus, for $V_2 = \{1,2\}$

$$\begin{aligned} d(3,V_2) &= \min\ (5,3) = 3 \;, \\ d(4,V_2) &= \min\ (3,4) = 3 \;, \\ d(5,V_2) &= \min\ (5,5) = 5 \;, \\ d(6,V_2) &= \min\ (4,6) = 4 \;; \end{aligned}$$

and so on for each possible V_2.

Second, find the radius associated with each V_2 where, as above, the radius is defined as the maximum of the set of minimum distances; i.e., the minimum distance to the farthest vertex from each V_2.

For $V_2 = \{1,2\}$ the radius $r(1,2)$ is

$$r(1,2) = \max_{v} d(v,V_2) = \max\ (3,3,5,4) = 5 \;.$$

Third, select the set of two vertices having the minimum radius; i.e.,

$$r_2 = r(V_2^*) = \min_{V_2 \subseteq V} r(V_2) , \qquad (6)$$

and specify that V which produces this minimum as V *. For the above example it may be readily verified that $r = 3$, with $V_2^* = \{1,5\}$.

In the general m-center case equation (6) becomes

$$r_m = \min_{V_m \subseteq V} \max_{v \varepsilon V} d(v,V_m) . \qquad (7)$$

2. The Absolute M-Center

The absolute m-center of G is defined as that set of m points X_m^* on G such that for every other such set of m points X_m on G,

$$r_{am} = \max_{v \varepsilon V} d(v,X_m^*) \leq \max_{v \varepsilon V} d(v,X_m) , \qquad (8)$$

where r_{am} is the absolute m-radius of G.

a. The Gillespie Algorithm

Gillespie [3] developed an algorithm for the best 2-center based on the vertex 2-center and Hakimi's algorithm for the absolute center of a graph. He partitions G into two subgraphs centered on the vertex 2-center, and by plotting the radius of a pair of points consisting of one member of the vertex 2-center and a moving point on an edge incident to the other member, points constituting the most centrally located 2-center are found.

Gillespie did not attempt to extend his solution technique to cases with $m > 2$, but did discuss the existence of the vertex and absolute m-centers and developed the following two theorems (Theorems 1 and 2, Ref. 3).

Theorem 3: Any graph containing at least m vertices has a vertex m-center.

Proof: This theorem is true from the definition of the vertex m-center. Since, for any graph with m vertices there will be at least one set V_m, there must exist a set V_m^*, the vertex m-center.

Theorem 4: Any graph having at least m vertices has an absolute m-center.

Proof: Theorem 4 follows directly from Theorem 3. Since V_m^* always exists and $V_m^* \subseteq X_m$, then X_m^* must also exist.

b. The Rosenthal-Smith Algorithm

In the extension of their algorithm for the absolute center of G to the absolute m-center, Rosenthal and Smith [7] partition G into m subgraphs G_i and apply their algorithm for the absolute center to each of these subgraphs. An iterative procedure is then used to compare the distance on G from each vertex v_j, $j = 1, \ldots, n$, to the absolute center x_i of each G_i, shifting vertices to different subgraphs when certain criteria are met, and recomputing the absolute center and radius of each affected subgraph. The algorithm terminates when no more vertices meet the criteria for being shifted.

As the basis for their method of partitioning G, these authors define the 'm-node divisional path P^m ', where m is the number of absolute centers desired, "...as that path which connects m nodes ($m \leq n$) such that the distance of the minimum branch connecting any two nodes in this path is greater than the minimum distance of the branch connecting any two nodes of any other path of G which connects m nodes." It should be observed that, as defined, the m-vertex (node) divisional path is actually a circuit when $m > 2$.

Let V_m be any set of m vertices in V, then P^m may be defined as that set V_m^* yielding

$$\max_{V_m \varepsilon V} \left[\min_{v \varepsilon V_m} d(v, V_m) \right] , \tag{9}$$

where $d(v_i, v_i) = 0$ is excluded from consideration. Since each vertex in the path can be represented by the two distances corresponding to the two edges of the path which are incident with it, (9) may be expanded in the following form (for four vertices);

$$\max_{V_4 \varepsilon V} \left\{ \min \left[d(v_r, v_s),\ d(v_s, v_t),\ d(v_t, v_u),\ d(v_u, v_r) \right] \right\} .$$

The procedure for finding P^m from the distance matrix is to find the m maximum $d_{ij} \varepsilon D$, $i < j$ (upper triangular portion of D), which form a circuit. Since all vertices of a cricuit are of degree two, no more than two entries may be taken from any row or column of D. The same problems arise here as in the procedure for finding the central path,

except here it must be verified that the route found is indeed a circuit (vice a path), with each edge traversed only once in completing the circuit.

III. THE VERTEX-M-CENTER APPROACH TO THE ABSOLUTE M-CENTER

In most of the simple examples included in previous works on the m-center problem, it appears that although they are seldom colocated, the absolute center is frequently located on an edge incident to the vertex center of the graph. Gillespie [3] commented on this, and his efforts were restricted to trying to find an absolute 2-center when it occurs on an edge incident to a vertex 2-center. The algorithm developed below uses this notion as the first step in searching for the absolute center, but goes beyond it and attempts to find the absolute center even when it is not on an edge incident to the vertex center.

A. DEFINITIONS AND CONVENTIONS

The following definitions and conventions are used in the development of the vertex-m-center-approach algorithm: Consider a graph G(V,A) with vertex center c and minimum distance tree Tc rooted on c; i.e., the tree such that d(v,c) is minimized for all $v \in V$. Similarly, Tv will denote the minimum distance tree rooted on any vertex $v \in V$.

Define 'cross-edge' to be any edge (i,j) in A but not in Tc such that, if (i,j) is added to Tc, a circuit (i,c,j,i) is produced.

Define 'extreme vertex' to be any vertex of degree one in Tc. Let the extreme vertices be ordered by distance from c and by the characteristics of the path to c from the extreme vertex and denoted p, q, r, ... or p_i, q_i, ... ,

$i = 1, 2, \ldots$, in accordance with the following conventions:

$$d(p,c) \geq d(q,c) \geq d(r,c) \geq \ldots \geq 0 \ .$$

There are no edges in common between the paths (v,c), $v = p, q, \ldots$; that is,

$$(p,c) \cap (q,c) \cap (r,c) \quad \ldots = \Phi \ .$$

The paths (v_1,c), (v_2,c), (v_3,c), ... have at least one edge in common with the path (v,c), $v = p, q, r, \ldots$;

$$\begin{array}{ccccccl} (p,c) & \cap & (p_1,c) & \cap & (p_2,c) & \ldots \neq & \Phi \ , \\ (q,c) & \cap & (q_1,c) & \cap & (q_2,c) & \ldots \neq & \Phi \ , \\ \vdots & & \vdots & & \vdots & \vdots & \vdots \ ; \end{array}$$

and they are ordered by length

$$\begin{array}{cccccccl} d(p,c) & \geq & d(p_1,c) & \geq & d(p_2,c) & \geq \ldots > & 0 \ , \\ d(q,c) & \geq & d(q_1,c) & \geq & d(q_2,c) & \geq \ldots > & 0 \ , \\ \vdots & & \vdots & & \vdots & \vdots & \vdots \ . \end{array}$$

Figure 2 shows a minimum distance tree rooted on the vertex center of a graph. Vertex 4 is the vertex center with a radius of ten; $d(4,10) = 10 = d(p,c)$. The extreme vertices are 1, 2, 3, 5, 8, 9, and 10; with corresponding distances to the vertex center $d(1,4) = 8$, $d(2,4) = 6$, $d(3,4) = 7$, $d(5,4) = 7$, $d(8,4) = 6$, $d(9,4) = 6$, and $d(10,4) = 10$. Since vertex 10 is the farthest from c (vertex 4),

it is designated p; vertex 1 is second farthest from c and is designated q, etc. Vertices 3 and 5 are equidistant from c, but the path from 3 to c has edge (4,7) in common with the path from vertex 10 (p), so vertex 3 is designated p_1 and vertex 5 becomes r by default. There is a three-way tie for s between vertices 2, 8, and 9. An arbitrary choice is made to assign s to vertex 2 and t to vertex 9, but then it is necessary to assign t_1 to vertex 8 because its path to c shares the edge (4,6) with the path from vertex 9 (t) to c.

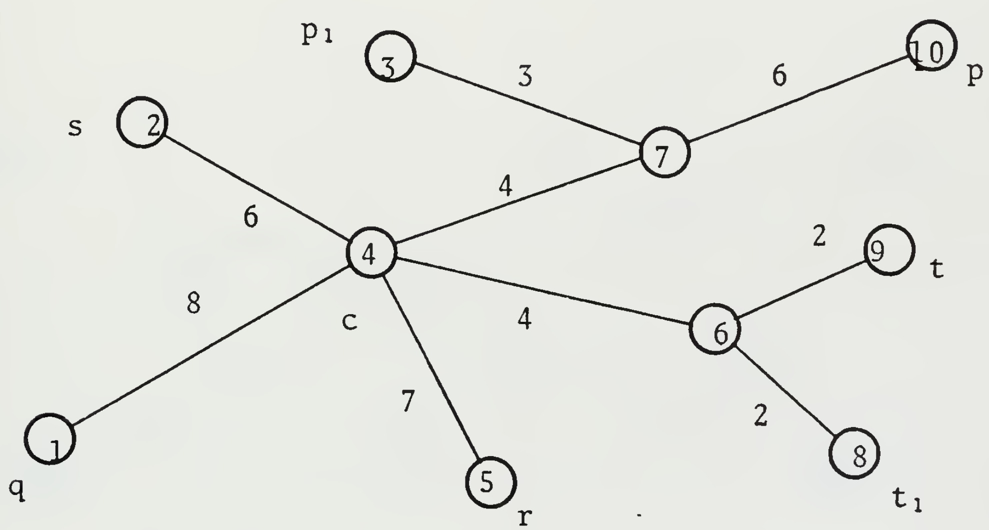


Figure 2. Minimum Distance Tree Rooted on the Vertex Center

B. AN UPPER BOUND ON THE ABSOLUTE RADIUS OF G

Clearly the absolute radius r_a of G satisfies $r_a \leq r_c$, for if x were a local center of G such that $r(x) > r_c$, then r(x) could be reduced by moving x to coincide with c, and x could not have been an absolute center of G.

Consider the point x_0 on the tree in Figure 3, where c is the vertex center and the extreme vertices are ordered in

accordance with the conventions outlined in the preceding section. Let x_0 be the midpoint of the path (p,c,q), then $d(x_0,p) = d(x_0,q) = r(x_0) = \frac{1}{2}[d(p,c) + d(q,c)]$. Since $d(q,c) \leq d(p,c)$, this implies that $r(x_0) \leq d(p,c) = r_c$, with equality holding when $d(q,c) = d(p,c)$. It can easily be shown that $d(x_0,p) \geq d(x_0,v)$ and $d(x_0,p) \geq d(x_0,v_i)$, $v = r, s, t, \ldots, i = 1, 2, \ldots$, with equality holding only if $d(v_i,c) = d(q,c)$, since $d(x_0,p) = d(x_0,q)$ by definition. That is, x_0 is no farther from any other extreme vertex than it is from p and q, the two most extreme vertices.

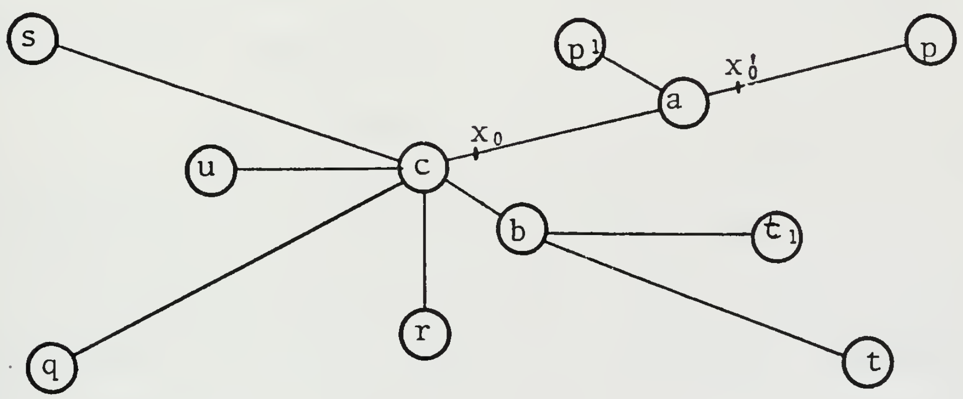


Figure 3: A General Minimum-Distance Tree

Now consider the point x_0', Figure 3, and let $d(p,x_0') \geq d(x_0',a) + d(a,p_1)$. If it is assumed that x_0' is an absolute center of G (with the vertex center still at c), then $d(p,x_0') = d(q,x_0')$ or

$$d(p,x_0') = d(x_0',a) + d(a,c) + d(c,q) .$$

Therefore

$$d(c,q) + d(c,a) < d(p,a) ;$$

but this is the condition necessary for vertex a to be the vertex center, which contradicts the original hypothesis that c is the vertex center of G. Note that the above analyses implicitly assume that the path (p,c,q) is the central path of G as defined by Rosenthal and Smith.

Hence it is concluded that the absolute center of G cannot occur on an edge in Tc which is not incident to the vertex center if that edge is an element of the central path. This does not preclude its occurrence on an edge of an alternate tree in case of ties for vertex center, nor on a cross-edge excluded from Tc.

Thus, the average of the two longest independent paths from the vertex center in Tc may be taken as an upper bound of the absolute radius of G:

$$r_a \leq \tfrac{1}{2}[d(p,c) + d(q,c)] = r(x_0) \ . \qquad (10)$$

C. MODIFYING Tc TO OBTAIN THE CENTRAL PATH

If the path (p,c,q) on Tc (as defined above) were the central path of G in all cases, the problem of locating the absolute center of G could be solved by merely applying equation (10). Unfortunately this does not hold true in many cases. If the central path passes through a vertex center of G (recall that the vertex center need not be unique) it is frequently possible to use Tc as a starting point in searching for the central path.

In general, when a cross-edge is added to Tc, alternate routes are formed between each pair of extremes vertices due

to the formation of a circuit. Thus, by inserting a cross-edge into Tc and removing an existing edge from Tc to break the circuit, a new tree Tc' is formed. If the longest path over Tc' is shorter than the path (p,c,q) over Tc then the midpoint of the newly defined path supercedes x_0, the midpoint of (p,c,q), as a candidate for the absolute center of G.

Figure 4 shows a general vertex-center tree Tc with representative cross-edges (shown in broken lines) which may be considered as alternate routes for the central path. The point x_0 is the midpoint of the path (p,c,q) as previously defined. If the path (p,i,h,q) over cross-edge (i,h) is shorter than (p,c,q), then adding (i,h) to Tc and removing either edge (i,c) or (h,c) to eliminate the circuit (i,c,h,i) may result in a reduction of the radius of G if the shorter of the pair of paths, (p,i,h,c,r) and (q,h,i,c,r) is retained. The radius of G will be reduced if this shorter path is also shorter than (p,c,q). For example, assume that (p,i,h,c,r) is shorter than (q,h,i,c,r) and that $d(p,c,q) > d(p,i,h,c,r) > d(p,i,h,q)$, and $d(p,i,h,c,r) \geq d(q,c,r)$. Then if edge (i,h) is inserted into Tc and (i,c) is removed, the path (p,i,h,c,r) becomes a new candidate for the central path of G with a radius less than $r(x_0)$.

Thus, if the cross-edge yielding the shortest candidate for the central path is found and inserted into Tc and the circuit is broken as outlined above, the absolute center will be found <u>in those cases where the central path passes through a vertex-center of G</u>. Since the vertex center need not be

unique, the minimum distance tree rooted on each of the alternate vertex centers must be inspected in turn. Restricting the algorithm to those cases where the central path does pass through a vertex center allows one to ignore the path lengths to any extreme vertices of higher order (closer to the vertex center) than the third most distant extreme vertex, r.

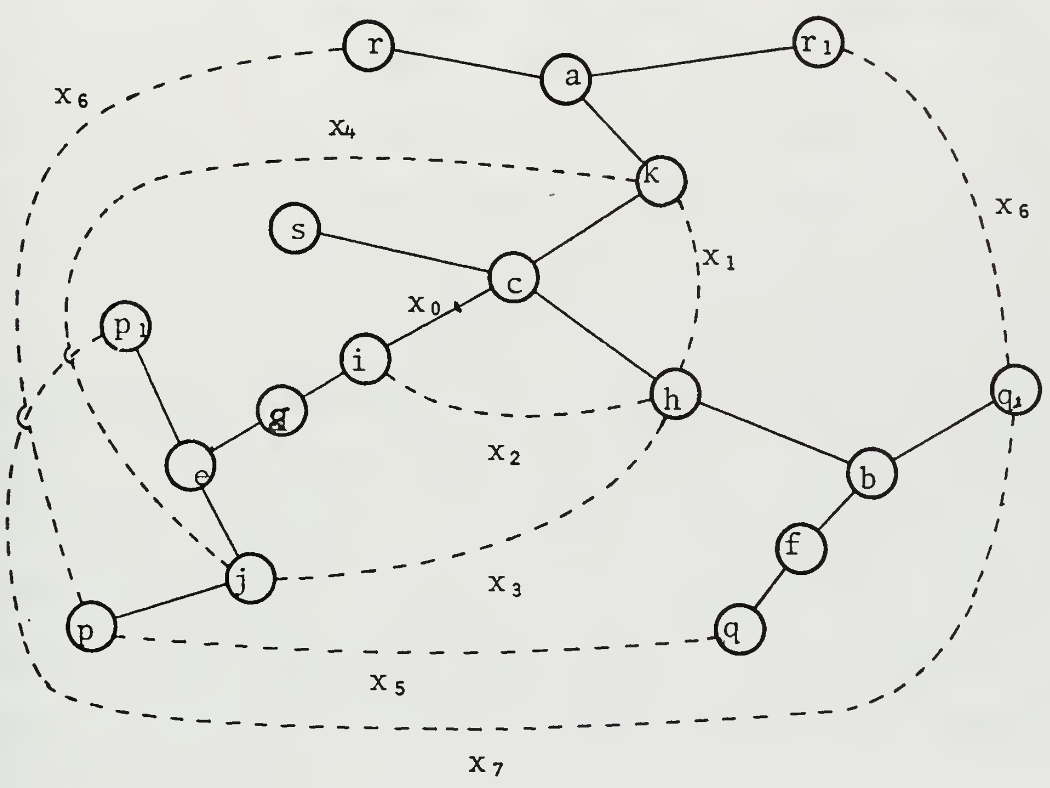


Figure 4: Some Sample Cross-Edges

To increase the generality of the discussions concerning cross-edges shown in Figure 4, it is assumed that there may exist additional vertices on any branch of Tc (solid line) in Figure 4 which are not shown in the figure. Therefore the distance between vertices which appear to be adjacent

in Tc as shown in Figure 4 will be referred to in distance notation rather than in edge-length notation; that is, the distance between vertices i and c will always be given as d(i,c) rather than as l(i,c) etc. There can be no intervening vertices on a cross-edge, so the length of each cross-edge will always be given in edge-length notation such as l(i,h).

D. CROSS EDGES ON THE CENTRAL PATH

The following paragraphs will develop the maximum allowable length for a cross-edge which will yield a reduction in the length of the central path if that cross-edge is inserted into Tc as outlined above. This is done for each of the representative cross-edges shown in Figure 4. The x_i, $i = 1, \ldots, 8$, shown in Figure 4 represent the midpoint of the central path candidate over the edge on which the respective x_i is located, where the candidate path is found as discussed in Section III.C above. Thus the point x_2 represents the midpoint of the path (p,i,h,c,r) under the conditions of the example discussed above. Later examples will show that this midpoint need not fall on the cross-edge under consideration.

First, consider the edge (k,h) with associated midpoint x_1 in Figure 4. If (k,h) is inserted into Tc and (c,k) is removed, the original central path candidate (p,c,q) with midpoint x_0 is not affected; thus no reduction in the radius of G is possible. If (c,h) is removed instead of (c,k), then d(c,k) + l(k,h) must be shorter than d(c,h) or the path

(p,c,k,h,q) will not be shorter than the original path (p,c,q). But if $d(c,k) + l(k,h) < d(c,h)$, the original path in Tc from the vertex center c to vertex h would have been (c,k,h) vice (c,h) by definition of a minimum distance tree rooted on c. This same result may be extended to edge (r_1,q_1) in Figure 4. Therefore it is concluded that no cross-edges between the paths (c,q) and (c,r) or (c,v), where $d(c,v) < d(c,q)$ and v is an extreme vertex, need be considered.

Now consider the cross-edge (i,h) in Figure 4. When (i,h) is inserted into Tc and the resulting circuit (i,c,h,i) is removed by breaking either of the paths (i,c) or (c,h), the new path between extreme vertices p and q is (p,i,h,q). The length of this path must not be greater than the original path between p and q, (p,c,q), or the radius of G will be increased; and it must be less than (p,c,q) to reduce the radius of G. In addition, the longer of the two paths (p,i,h,c,r) and (q,h,i,c,r) can be removed by breaking either (c,h) or (c,i) respectively. Therefore the shorter of these two paths will remain and must also be shorter than (p,c,q) to allow a reduction in the radius of G. These conditions are equivalent to

$$l(i,h) \leq d(i,c,h) = d(i,c) + d(c,h) \qquad (11)$$

and

$$\min \left\{ \begin{array}{l} d(p,i) + l(i,h) + d(h,c,r) \\ \\ d(q,h) + l(i,h) + d(i,c,r) \end{array} \right\} \leq d(p,c,q). \qquad (12)$$

Inequality (12) may be rewritten in the same form as (11),

$$l(i,h) \leq \max \begin{cases} d(i,c) + d(h,q) - d(c,r) \\ \\ d(h,c) + d(i,p) - d(c,r) \end{cases} \quad .(12a)$$

Therefore, if l(i,h) satisfies inequalities (11) and (12), cross-edge (i,h) may be inserted into Tc, the resulting circuit removed by breaking the longer of the two paths (p,i,h,c,r) and (q,h,i,c,r) between c and i or h, and a new candidate for the central path will be obtained. If strict inequality holds in both (11) and (12), this will yield a new radius of G which is strictly less than $r(x_0)$.

Next, consider the cross-edge (j,h) in Figure 4; (j,h) differs from (i,h) in that the secondary extreme vertex p_1 is connected to the path (p,c) at a point (vertex e) which is closer to the vertex center than is vertex j. It is therefore possible for the path (p_1,e,j,h,c,r) to be longer than the path (p,j,h,c,r), and inequality (12) must be modified to hold for the longer of these two paths. When this is done, (12) becomes

$$l(i,h) \leq \max \begin{cases} \begin{bmatrix} d(p,j,e,c) + d(h,q) - d(c,r) \\ -\max\,[d(p_1,e,j);\, d(p,j)] \end{bmatrix} \\ d(p,j) + d(h,c) - d(c,r) \end{cases} \quad (12b)$$

Inequality (11) remains in the same form, with j substituted for i;

$$l(j,h) \leq d(j,c,h) = d(j,c) + d(c,h) \qquad (11a)$$

The roles of p and p_1 may be interchanged in this case, and similar branching may occur in paths associated with

with vertex f such as over cross-edges (i,f) or (j,f) (not shown in Figure 4). These situations and role interchanges between q and q_1 would still be handled in a similar manner but modifying (11) and (12) as was done in (11a) and (12b).

The same procedures also apply for the cross-edge (j,k) in Figure 4. In this case the roles of q and r are interchanged in (12a), and (11) is modified also.

To consider the cross-edges (p,q) and (p,r) in Figure 4, the simplified graph in Figure 5 will be useful. For cross-edge (p,q), in the absence of secondary extreme vertices as shown in Figure 5, the appropriate form of the path length criterion (12) becomes

$$\text{(a)} \quad \min \left\{ \begin{array}{l} l(p,q) + d(q,c) + d(c,r) \\ l(p,q) + d(p,c) + d(c,r) \end{array} \right\} \le d(p,c) + d(q,c) \ ,$$

and the appropriate form of (11) is

$$\text{(b)} \quad l(p,q) \le d(p,c) + d(q,c) \ .$$

But (b) is less restrictive than (a), and so can be discarded. Then inequality (a) may be re-written in the form

$$(a_1) \quad l(q,p) + d(c,r) \le \max \left\{ \begin{array}{l} d(p,c) \\ d(q,c) \end{array} \right\} .$$

But by definition, $d(p,c) \ge d(q,c)$; therefore (a_1) becomes

$$l(p,q) \le d(p,c) - d(c,r) \quad . \qquad \text{(12c)}$$

For the cross-edge (p,r), in the absence of secondary extreme vertices as in Figure 5, the appropriate form of path length criterion (12) is

(b)
$$\min \begin{Bmatrix} l(p,r) + d(c,r) + d(c,q) \\ l(p,r) + d(c,p) + d(c,q) \end{Bmatrix} \le d(c,p) + d(c,q) \;;$$

and since $d(c,p) - d(c,r) \geq d(c,p) - d(c,q)$ by definition, this reduces to

$$l(p,r) \le d(c,p) - d(c,r) \quad . \tag{12d}$$

Notice that the right sides of (12c) and (12d) are identical. This is due to the simple role interchange of q and r in the left side of (a) to arrive at (b), and illustrates the ease with which the situations discussed on the preceding page may be handled. Once again the proper form of (11) is less restrictive than (b) and may be ignored, but in this case (12d) _is_ the appropriate form of (11). When secondary vertices are involved, the appropriate form of (12) is not so simple as (12c) and (12d), as is shown below.

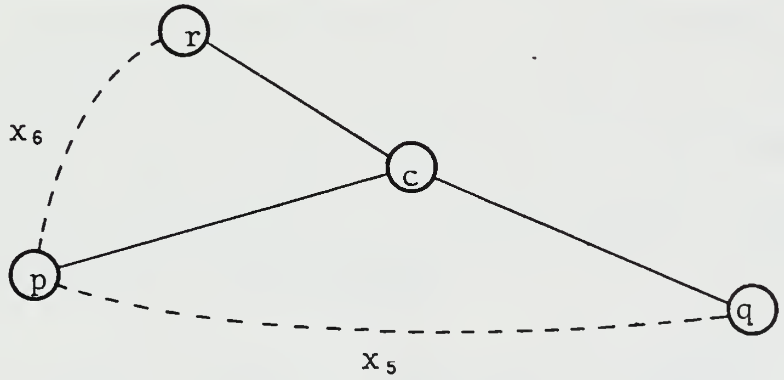


Figure 5: Simplification of Figure 4.

Now return to Figure 4 and reconsider cross-edge (p,q). In the presence of secondary extreme vertices, the proper

form of distance criterion (12) is

(c)

$$\min\begin{Bmatrix} d(p_1,e,p) + l(p,q) + d(q,c,r) \\ \\ d(q_1,b,q) + l(p,q) + d(p,c,r) \end{Bmatrix} \leq d(p,c,q) \;;$$

If $d(p_1,e,p) = 0$ and $d(q_1,b,q) = 0$, (c) reduces to (12c) as would be expected. As in the previous case, (c) is more restrictive than the appropriate form of (11), and is equivalent to

$$l(p,q) \leq \max\begin{Bmatrix} d(e,c) - d(p_1,e) \\ \\ d(b,c) - d(q_1,b) \end{Bmatrix} - d(c,r) \;. \qquad (12e)$$

From (c) one may also deduce the further dual requirements that

$$\begin{aligned} d(p,c) &> d(p_1,c) \;, \\ d(q,c) &> d(q_1,c) \;. \end{aligned} \qquad (13)$$

By interchanging the roles of p and p_1 and of q and q_1, inequality (c) becomes applicable to a cross-edge between secondary vertices such as (p_1,q_1) in Figure 4. In this case the proper form of (11) is no longer less restrictive than the revised version of (c), and so must be considered again. Relation (11) now becomes

$$l(p_1,q_1) \leq [d(e,c) - d(e,p_1)] + [d(b,c) - d(b,q_1)]. \qquad (11b)$$

While the correct form of (c) is

$$l(p_1,q_1) \leq \max\begin{Bmatrix} [d(e,c) - d(e,p_1)]+[d(b,q) - d(b,q_1)] \\ \\ [d(b,c) - d(b,q_1)]+[d(e,p) - d(e,p_1)] \end{Bmatrix} - d(c,r). \qquad (12f)$$

Throughout the preceding discussion, one important factor has been ignored, primarily because it is impossible to quantify in a general sense. When a cross-edge has been added to Tc and an edge of Tc is removed to break the resultant circuit, additional extreme vertices which were not extreme in Tc may be generated in the modified tree. For instance, if cross-edge (j,h) in Figure 4 has been found to satisfy (11a) and (12b) and is added to the tree in Figure 4, when the circuit (j,e,g,i,c,h,j) is broken by removing (g,i) (assuming path (g,i) to now be a single edge), vertices g and i become newly defined extreme vertices. It is then possible for the path (i,c,h,j,p) or (g,e,j,h,c,r) to be longer than the initial controlling path, (p,c,q). This possibility must therefore be avoided in all instances. Example 4, Section III.F will illustrate a case in which it is not possible to modify Tc because of the occurrence of newly defined extreme vertices with associated path lengths greater than the length of (p,c,q).

E. THE ALGORITHM

The discussion of the preceding sections may be summarized in step-wise manner as follows:

1. Find the vertex center of the graph. See Refs. [3] and [4] or Section II above.

2. Form the minimal distance tree Tc rooted on the vertex center c using the Dijkstra Algorithm or a similar method (see Dreyfus [1]).

3. Order the extreme vertices as outlined in Section III.A.

4. Find the midpoint x_0 of the longest path (p,c,q) over Tc.

5. Check all cross-edges from vertices in the path from p to c with the applicable form of inequalities (11) and (12).

6. Choose the cross-edge which yields the shortest central path.

7. Insert that cross-edge into Tc and break the resulting circuit so that the longest non-recursive path between a pair of extreme vertices is removed. Break the circuit so as to minimize the length of the longest such path from a newly defined extreme vertex.

8. If the longest path from a newly defined extreme vertex is longer than path (p,c,q), drop this cross-edge from consideration, return to step 6 to find the cross-edge yielding the next shortest central path candidate.

9. If the longest path from a newly defined extreme vertex is shorter than (p,c,q) but longer than the longest path defined by the applicable forms of (11) and (12), retain the result but return to 6 to find the cross-edge yielding the next longer central path candidate.

10. Find the midpoint of the shortest central path candidate (longest remaining path between any pair of extreme vertices including newly defined extreme vertices) from all iterations of steps 6 through 9.

11. Repeat steps 2 through 10 for each alternative vertex center.

12. Choose the local center from step 10 having the minimum radius as absolute center of the graph.

F. EXAMPLES

The following two examples will illustrate the application of the vertex-center-approach algorithm. Example 3 illustrates a case in which the central path is found to contain a cross-edge, while Example 4 illustrates how newly-defined extreme vertices can prevent this.

Figure 6, Example 3, shows a graph with the edges in Tc in solid lines and the cross-edge as a broken line. The extreme vertices and vertex center are appropriately marked. The distances between all pairs of extreme vertices, including non-recursive paths over cross-edge (4,5), are

$$
\begin{aligned}
d(p,c,q) &= d(3,4,1,5,6) = 35 \quad , \\
d(p,c,r) &= d(3,4,1,2) = 24 \quad , \\
d(q,c,r) &= d(6,5,1,2) = 23 \quad , \\
d(p,4,5,q) &= d(3,4,5,6) = 18 \quad , \\
d(p,4,5,c,r) &= d(3,4,5,1,2) = 33 \quad , \\
d(q,5,4,c,r) &= d(6,5,4,1,2) = 34 \quad .
\end{aligned}
$$

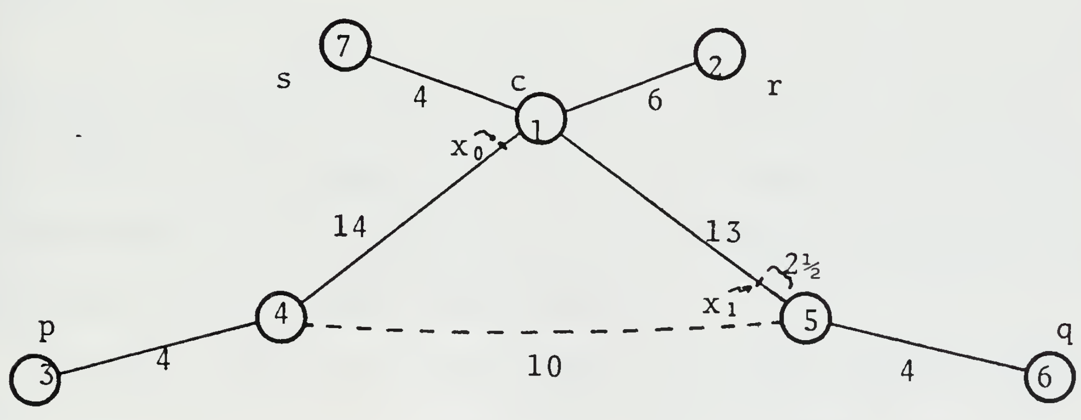


Figure 6: Example 3

It is clear from this list of path lengths that <u>all</u> alternate paths are shorter than (p,c,q). Since removing either edge (1,4) or (1,5) breaks (p,c,q), it is ignored and only paths (p,4,5,c,r) and (q,5,4,c,r) need be considered, as outlined in Section III.D, since these are the two next longest paths. The longer of these two is the path (q,5,4,c,r) with a path length of 34, and it can be broken by removing edge (1,4) in the circuit (1,5,4,1). Thus, removal of edge (1,4) eliminates both paths (p,c,q) and (q,5,4,c,r), and leaves path (p,4,5,c,r) as the longest path over the modified tree, with a length of 33. The midpoint of this path is the point x_1 with a radius $r(x_1) = 33/2 = 16\frac{1}{2}$. The reader may readily verify that x_1 is the absolute center of the graph in Figure 6 by applying the Rosenthal-Smith algorithm.

The use of relations (11) and (12) were omitted above for illustrative purposes. They would normally be applied in step 5 of the algorithm in the following manner:

$$l(4,5) = 10 \le d(4,1,5) = 27 \; ; \qquad (11)$$

$$l(4,5) \le \max \left\{ \begin{array}{ll} d(4,1) + d(5,4) - d(1,2) = 14 + 4 - 6 = 12 & (12a) \\ d(5,1) + d(3,4) - d(1,2) = 13 + 4 - 6 = 11 & \end{array} \right\} = 12.$$

Since both relations hold for strict inequality, one knows that adding cross-edge (4,5) to Tc will result in a reduction of the radius of G providing no newly-defined extreme vertex generates a path longer than (p,c,q).

Had the length of (4,5) been 12 so that equality held in (12a), path (p,4,5,c,r) would have been 35 units long, the

same as (p,c,q). In this case x_1 would be an alternate absolute center, with the point x_0, the midpoint of (p,c,q), being the other; and both would have a radius of 35/2 = 17½.

If l(4,5) is increased to greater than 12, both paths (p,4,5,c,r) and (q,5,4,c,r) become longer than (p,c,q); and no reduction in the radius of G is possible. The point x_0 is the only absolute center in this case.

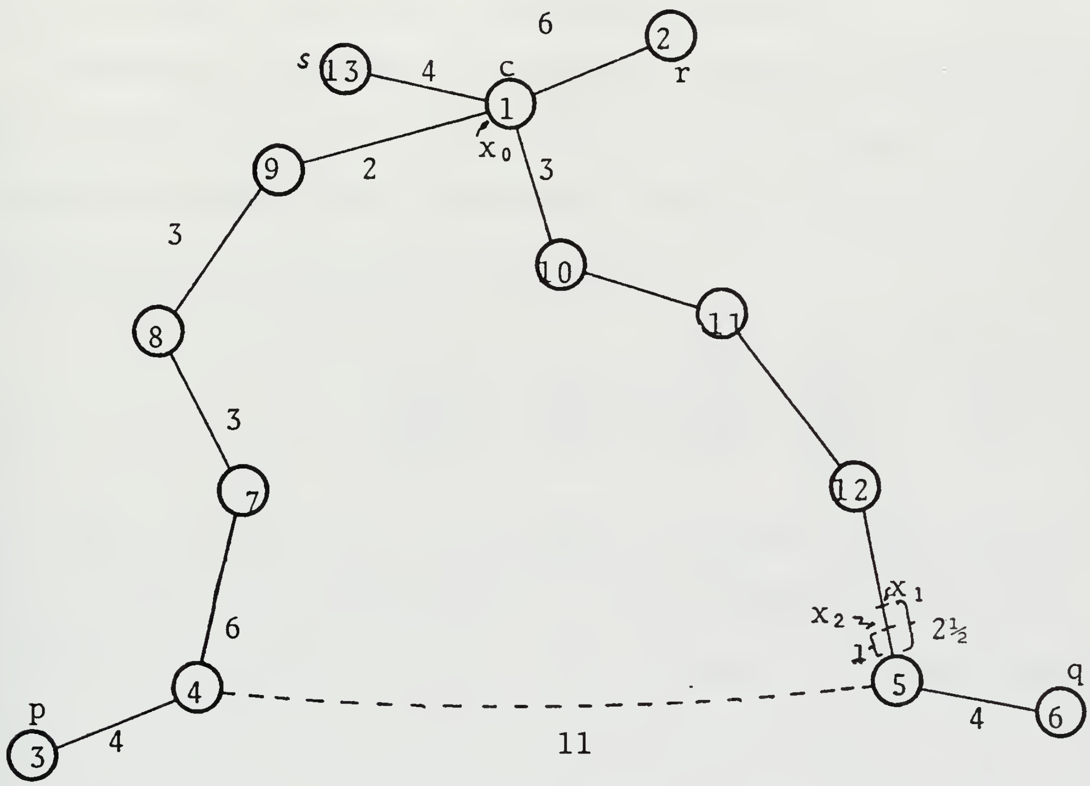


Figure 7: Example 4.

Figure 7, Example 4, is a modification of Figure 6 with vertices 7 through 12 added. The distance d(1,5) is increased to 14 units from 13 and l(4,5) is increased to 11 units. Relations (11) and (12a) are still the applicable forms of the distance criteria, but now they are

$$l(4,5) = 11 \leq d(4,1,5) = 28 \tag{11}$$

$$l(4,5) \leq \max \left\{ \begin{array}{l} 14 + 4 - 6 = 12 \\ 14 + 4 - 6 = 12 \end{array} \right\} = 12. \tag{12a}$$

This indicates that adding cross-edge (4,5) to Tc and breaking the circuit along either path (4,1) or (5,1) will reduce the radius of G. Since d(q,5,4,c,r) = d(p,4,5,c,r) = 35 and d(p,c,q) = 36, it would seem that either (4,1) or (5,1) could be broken and obtain equal reductions.

Arbitrarily choose to break the circuit by removing an edge in the path (4,7,8,9,c). When each of these four edges is removed in turn, the following distances from newly defined extreme vertices are obtained:

by removing	(4,7),	d(7,c,5,4,p) = 40	<	d(p,c,q);
" "	(7,8),	d(8,c,5,4,7) = 36	=	" ;
" "	(8,9),	d(9,c,5,4,8) = 36	=	" ;
" "	(9,c),	d(r,c,5,4,9) = 43	<	" .

From these results it is seen that if either edge (4,7) or (9,c) is removed, the radius of G is <u>increased</u>. If (7,8) is removed, then $d(7,4,5,x_1) = 17\frac{1}{2}$, where x_1 is the midpoint of the path (p,4,5,c,r) with $d(p,4,5,x_1) = d(r,c,12,x_1) = 17$. Since the midpoint x_0 of the path (p,c,q) is colocated with the vertex center and has radius $r(x_0) = r_c = 18$, x_1 may be moved ½ unit in either direction before its radius becomes greater than $r(x_0)$. However, moving x_1 ½ unit toward vertex 5 reduces $d(7,4,5,x_1)$ to only 19 units which is still greater than $r(x_0)$. Similar results occur when edge (8,9) is removed and when one attempts to break the circuit by removing an edge in the path (c,10,11,12,5).

Thus no reduction in the radius of the graph in Figure 6 is possible due to the generation of newly-defined extreme vertices, and it is concluded that the point x_0 is the absolute center of the graph with $r_a = r_c = 18$.

In both of the above examples the vertex center of the example graph has been unique. When there are alternate vertex centers, the algorithm is repeated for each vertex center as given in step 11 of the algorithm, and the optimum result(s) is(are) chosen in step 12.

The reader should bear in mind that this algorithm will find the absolute center of a graph only when the vertex center lies on the central path of G.

G. EXTENSION TO THE ABSOLUTE M-CENTER

The algorithm may be extended to deal with the absolute m-center, m > 1, in the same manner used by Gillespie.

1. Find the vertex m-center, $V_m{}^* = \{c_i\}$, $i = 1, \ldots, m$.

2. Form m minimum distance trees, Tc_i rooted on c_i, $i = 1, \ldots, m$, by connecting each vertex in V to the nearest c_i. If any vertex is equidistant between two or more vertex centers, place it in both (all) associated trees.

3. Apply the preceding algorithm for the absolute center to each of the trees Tc_i, $i = 1, \ldots, m$. Do not consider any cross-edges between vertices not in the same tree.

4. If any vertex was found to be equidistant between two or more of the m vertex centers in step 2, choose that affected tree having the smallest individual absolute radius

to retain the tying vertex, remove it from the other trees and repeat step 3 for those affected trees.

5. Choose the maximum of the set of m individual absolute radii as the absolute m-radius r_{am} for this partition of G.

6. Repeat steps 1 through 5 for all alternate vertex m-centers.

7. Choose that vertex m-center (partition of G) yielding the minimum absolute m-radius as the final solution.

Gillespie used the vertex 2-center as the basis for partitioning a graph in the first step of his algorithm, but avoided any consideration of its validity or uniqueness other than to examine alternate vertex 2-centers. The extension of this method to the use of the vertex m-centers is used here with the implicit understanding that, while the result it yields may not be optimal, it at least gives a relatively easily determined upper bound for the problem when m is small. While no research has been done on the subject, it is believed that the results become less reliable as m increases. No examples will be given of the application of the above algorithm.

Rosenthal and Smith [7] make no mention of alternate optima; and whereas their method of moving vertices into different subgraphs until no further reduction in radii is achieved apparently does find an optimum solution, it is not clear what is done when alternate optimum centers exist. Their method appears to minimize <u>all individual</u> radii to the

greatest extent possible rather than concentrating on the maximum radius, which would satisfy the definition of the absolute m-radius given herein.

IV. THE ARC-CENTER APPROACH TO THE ABSOLUTE M-CENTER

Recall that by Theorem 2 the absolute center of graph G is located at the midpoint of the central path, where the central path has been defined as the path (v_x, v_y, v_z) satisfying

$$d(v_x, v_y, v_z) = \min_j \left[\max_{i,k} d(v_i, v_j, v_k) \right] .$$

The Rosenthal-Smith algorithm based on this definition proves to be an iterative process which seldom finds a non-recursive path in the first iteration. It was felt that one of the primary shortcomings of this method of attack is the lack of any definite relationship between the location of the absolute center and the middle vertex of the trio defining the central path. That is, the absolute center need not occur on an edge incident to the middle vertex. There may in fact be several choices for the middle vertex, all defining the same path.

Since the absolute center of G is constrained to lie at some point on G, and therefore lies either at an interior point or end point (vertex) of some edge in G, it was decided that a procedure based on finding a central arc or arc-center (an edge containing an absolute center of G) might be more efficient. This led to the arc-center algorithm for the absolute center of a graph, developed in the following paragraphs. Since the absolute center need not be unique, the arc-center need not be unique.

A. FINDING THE ARC-CENTER

Define the arc-center to be an edge in G containing an absolute center of G. It should be noted that, in the event an absolute center lies at a vertex of G and only in that event, there are two arc-centers (or equivalently, none) associated with that particular absolute center.

Let the edge (i,j) be the arc-center of G and partition G into two subgraphs $G_i(V_i,A_i)$ and $G_j(V_j,A_j)$ such that all elements of V_i are closer to vertex i than to vertex j and all elements of V_j are closer to vertex j than to vertex i as determined from the distance matrix D (See Figure 8). There may be vertices which are equidistant between i and j and thus could be placed in either G_i or G_j. This event will be referred to as a type A tie, and a method of dealing with it will be developed later; but for the time being, assume that no type A ties exist.

Let the vertices in V_k be ordered according to distance from vertex k in a manner similar to that developed in Section III, and denote them p_k, p_{k1}, p_{k2}, ... , q_k, q_{k1}, ... , r_k, ... , etc. For brevity let their corresponding distances to vertex k be denoted $d(p_k,k) = dp_k$, $d(p_{k1},k) = dp_{k1}$, ... Then, associated with edge (i,j),

$$dp_i = d(p_i,i) = \max_i \left\{ \min_{v \in V} \left[d(v,i),\ d(v,j) \right] \right\} = \max_{v \in V_i} d(v,i) \ ; \tag{14}$$

$$dp_j = d(p_j,j) = \max_j \left\{ \min_{v \in V} \left[d(v,i),\ d(v,j) \right] \right\} = \max_{v \in V_j} d(v,j) \ .$$

Ties between extreme vertices within G_i and G_j may be resolved arbitrarily.

Using this notation, a <u>nearly</u> equivalent definition of the central path is that path P* in G(V,A) which yields

$$l(P^*) = \min_{(i,j)\varepsilon A} [\, dp_i + dp_j + l(i,j)]. \qquad (14a)$$

If an absolute center x* occurs on any edge (i,j) then $dp_i + d(i,x^*) = dp_j + d(j,x^*)$; or equivalently, $d(j,x^*) - d(i,x^*) = d(p_i,i) - d(p_j,j)$. If the absolute values of the latter equation are taken, one obtains

$$l(i,j) \geq |dp_i - dp_j| = |d(p_i,i) - d(p_j,j)|, \qquad (15)$$

since

$$l(i,j) \geq |d(j,x^*) - d(i,x^*)|.$$

On first consideration it may appear that if all M possible partitions of G into two subgraphs (as described above) for each edge (i,j) in A were examined and $l(P) = [dp_i + dp_j + l(i,j)]$ was computed for each partition, an edge which simultaneously satisfies (14a) and (15) would necessarily be found. The absolute center x* would then be the midpoint of the path $P^* = (p_i,i,j,p_j)$, and it would be located on edge (i,j) a distance

$$d(i,x^*) = \tfrac{1}{2}\,[l(i,j) - dp_i + dp_j] \qquad (16)$$

from vertex i. However, due to the existence of circuits in G, it frequently happens that one or more vertices are placed in G_i but would be closer to an absolute center if they had been placed in G_j. When this occurs, incorrect path lengths are defined.

For instance, in Example 6, Figure 10, the absolute center is colocated with vertex 1, and the shortest path from vertex 5 to the absolute center is along edge (1,5) with a length of five units. But vertex 5 is only two units from vertex 4 along edge (4,5), so when set (1,4) is tabulated, vertex 5 is placed in V_4 instead of V_1 which makes $d(5,1) = 7$ in this partition. Furthermore, since vertex 5 is the primary vertex p_4 in V_4, an incorrect central path candidate is defined: Path (5,4,1,3). Had vertex 5 been placed in V_1, the central path candidate would have been the path (5,1,4) which is the actual central path of the graph. Note that $d(5,4,1,3) = 9 < d(5,1,4) = 10$.

It is generally true that, if any vertex is incorrectly partitioned and would have been a primary (most extreme) vertex under correct partitioning, an incorrect central path candidate is defined which is shorter than the actual central path of the graph. A method of dealing with these cases of incorrect partitioning is developed below by using the approach developed by Rosenthal and Smith in defining the central path.

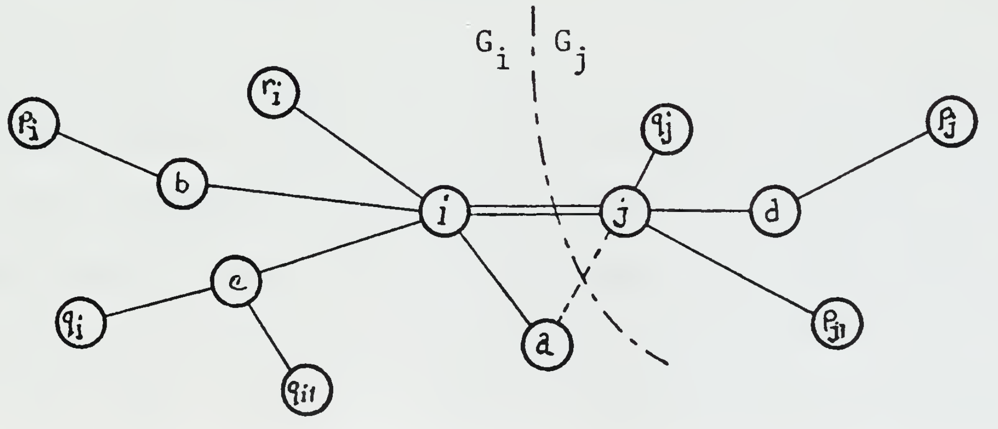


Figure 8: A Representative Partition of G.

Consider the representative partition of G into $\{G_i, G_j\}$ shown in Figure 8 and note that there are three possible candidates for the central path shown in the figure. These are the paths (p_i, i, j, p_j), (p_i, i, q_i), and (p_j, j, q_j). If the path (p_i, i, j, p_j) is the actual central path of G, one of two possible conditions will exist; either the absolute center is on (i,j) (in which case (15) will be satisfied by (i,j)) or the absolute center is on another edge such as (b,i) or (j,d) (in which case (i,j) cannot satisfy (15)). In either case, however, the path (p_i, i, j, p_j) must satisfy (14a). In the latter case it is impossible to predict what conditions will arise when G is partitioned into $\{G_b, G_i\}$ and $\{G_j, G_d\}$ without knowing what the connective pattern and distances are in G, but it is evident from Figure 8 that the set G as it occurs in $\{G_i, G_j\}$ differs from the set G_i that would occur in $\{G_b, G_i\}$. It will become apparent in later examples that this is generally true.

Now assume that $d(p_i, i, q_i) > d(p_i, i, j, p_j)$ in Figure 8. This is equivalent to

$$d(q_i, i) > d(p_j, j) + l(i,j) \ . \tag{17}$$

When this occurs, edge (i,j) cannot satisfy (15) because (17) must also hold if $d(p_i, i)$ is substituted for $d(q_i, i)$ in (17); but this requires the midpoint of the path (p_i, i, j, p_j) to lie on the branch (p_i, i), not on the edge (i,j). When the distances dp_i, dq_i, and dp_j for a given partition of G are such that (17) holds, this will be referred to as a 'B condition'.

Thus, when a B condition occurs, the path (p_i,i,q_i) is the longest non-recursive path through vertex i, and so becomes a primary candidate for the central path of G as defined by Rosenthal and Smith [7]. Therefore, if (p_i,i,q_i) is the shortest path found, it is the cental path of G, and the absolute center of G is the midpoint of (p_i,i,q_i) with absolute radius

$$r_a = \tfrac{1}{2}[d(p_i,i) + d(q_i,i)] .$$

Vertex a in Figure 8 is intended to be equidistant between vertices i and j; a type A tie. Under the condition of the figure this creates no problem, but had vertex a been either p_i or p_j the paths (a,i,j,p_j) with a in G_i and (p_i,i,j,a) with a in G_i (assuming a can be either p_i or p_j) would probably be of different lengths. Should this occur, both paths must be checked and the shorter chosen as the central path candidate; since by definition, if the central path is either of these paths, it must be the shorter of the two.

It is also possible that (i,j) does satisfy (15) when vertex a is in G_i, but does not when a is in G_j; or that a enters into a B condition as p_i or q_i when in G_i; but (15) is satisfied when a is in G_j, etc. The algorithm checks all possible combinations of type A ties and B conditions, and selects the shortest central path candidate as the central path of G.

B. THE ALGORITHM

Using the results of the preceding discussion, the arc-center algorithm for the absolute center of G may be formulated as follows.

1. Find the minimum of d(v,i) and d(v,j) for all M pairs of adjacent vertices by working from the distance matrix D using the same technique as for the vertex 2-center. This gives M different partitions $\{G_i, G_j\}$ of G. When doing hand calculations, the results are best tabulated in an (n+5) x (M+1) tableau as follows (see later examples, also).

a. List all edges (pairs of adjacent vertices) of G in row 1, columns 2 through (M+1); i.e., across the top of the tableau.

b. List all vertices in G in column 1, rows 2 through (n+1).

c. Place the following labels in column 1 rows (n+2) through (n+5) respectively: 'dp_i, dp_j', 'l(i,j)', 'l(P)', 'B'.

d. In row (n+3) (labeled 'l(i,j)') write the length of each edge in the column headed by the pair of adjacent vertices defining that edge.

e. Divide the column under each pair of vertices into two sub-columns in rows 2 through (n+1). Apply the vertex 2-center algorithm to G for each pair of vertices in row 1. Enter the minimum value of [d(v,i), d(v,j)] for each vertex v in the row labeled with that vertex and the sub-column headed by the vertex i or j for which the minimum occurs

(the cell of the tableau corresponding to v and i or j, whichever minimizes [d(v,i), d(v,j)]). If d(v,i) = d(v,j), enter this value in both subcolumn i and subcolumn j and mark this as defining a type A tie by placing 'A' in a convenient location, such as between the two entries.

f. When completed with step e, examine the minimum distance entries in each set (column) and, if the maximum entry in one subcolumn is less than or equal to the second or higher ordered (lesser valued) maximum entry in the other subcolumn of the set, mark the set as having a possible B condition by placing 'b' in a convenient location, such as beside the smallest entry which is greater than or equal to the maximum entry in the other subcolumn.

2. For each partition (set or column) find the $\max_{v \varepsilon V_i} d(v,i) = dp_i$ and the $\max_{v \varepsilon V_j} d(v,j) = dp_j$.

a. If a set has been marked as having a type A tie and the tied value is either dp_i or dp_j, delete the tying value from G_j and find the resulting values of dp_i and dp_j, then move the tying value from G_i to G_j and recompute dp_i and dp_j. This defines two different sets $\{G_i, G_j\}$ wherein the vertex having the tied value is in G_i for one set, and in G_j for the other.

b. Should there be more than a single pair of type A ties, say k pairs, in a given set, each of the 2k possible significant combinations is tried as follows:

1) The maximum tying value is deleted from G_j; all others remain in G_i. Check that the resulting route is

a non-recursive path and if it is, complete the calculations. If it is a recursive path, the tied values now in G_i are moved to G_j sequentially, and the route is checked for a non-recursive path each time; only the combination which determines the longest path is retained, all others being dropped from consideration since this is part of the maximizing step in finding the central path.

2) When step 1) is completed, move the maximum tied value to G_j and repeat step 1), moving the others to G_i sequentially. When this process is completed, select the non-recursive path(s) with the maximum length $dp_i + dp_j + l(i,j)$, including ties, and use it (them) as the path(s) representing the set(s). Ties yield alternate paths and, in the event their path lengths are the minimum of all central path candidates, they will yield alternate absolute centers.

3. For each partition $\{G_i, G_j\}$ compute $l(P) = dp_i + dp_j + l(i,j)$ to find the total path length.

4. Choose the minimum value in the set of total path lengths. If this is greater than the length of any legitimate central path candidate previously resolved, go to step 8 and terminate. If not, test the set with the distance criterion (15).

5. a. Go to step 6 with any set(s) (including ties) which satisfy (15).

b. If the minimum length path found in step 4 fails to satisfy (15), check for a B condition. In filling in the tableau <u>possible</u> B conditions are marked; i.e., if

the second maximum entry is subcolumn i is greater than the maximum entry in subcolumn j, the set is marked as having a possible B condition. Let dk_1 be the maximum entry in subcolumn k, dk_2 the second maximum entry, etc., k = i, j. Then if $di_2 > dj_1 + l(i,j)$ it is possible that a B condition exists, but it is necessary to check that a non-recursive path exists from the vertex yielding di_1 to vertex i and then to the vertex yielding di_2. Note that dk_1 is always dp_k, k = i, j (in step 2.b. only the maximum value was considered in each subcolumn, so it was not necessary to consider this problem at that time). When the value of dk_h, h = 2, 3, ... , has been found which yields the longest non-recursive path (p_k,k,q_k) (where q_k is the vertex yielding this value of dk_h), check that this dk_h still satisfies (17); i.e., for k = i, check that $di_h = dq_i = d(q_i,i) \geq d(p_j,j) + l(i,j)$.

1) If this holds, a B condition exists; the local radius is $\frac{1}{2}(dp_k + dq_k)$ and the local center is located at the midpoint of the path (p_k,k,q_k) where k refers to the subset G_i or G_j containing the B condition. Enter the sum $(dp_k + dq_k)$ in the 'B' row of the tableau. Return to step 4 and choose the next minimum total path length. If this is longer than the B condition path $(dp_k + dq_k)$ just found, go to step 8 with the results from the B condition set just resolved and terminate.

2) If $dq_i < dp_j + l(i,j)$, discard the set; return to step 4 and choose the set having the next minimum $l(P)$.

6. Divide the total minimum path length by two to find the local radius.

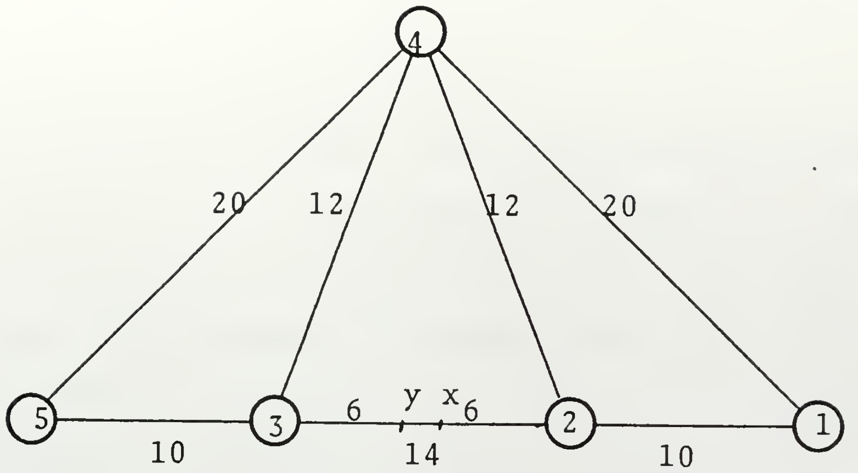
7. If no condition B was involved, locate the local center x^* at a distance $d(i,x^*)$ units from vertex i along (i,j) where $d(i,x^*)$ is as defined in equation (16). If a B condition was involved, the local center is found as given in step 5.b.1).

8. The algorithm terminates when the set with minimum total path length satisfies (15), or when it is verified that the condition B path of minimum length is found and is shorter than or equal to the shortest path satisfying (15).

C. EXAMPLES

Hakimi's example (Figure 1) will be used first to illustrate the application of the algorithm. The graph in Figure 1 is reproduced in Figure 9 for convenience, along with the associated distance matrix and the computational tableau resulting from applying the arc-center algorithm to locate the absolute center of this graph.

Upon completion of steps 1, 2, and 3 of the algorithm, the tableau appears as in Figure 9 with the exception that the 'B' row is not filled in. The only type A tie involved occurs in set (2,3) for vertex 4. Step 2.a. resolves the tie by first placing vertex 4 in G_2 (deleting the entry '12' from subcolumn 3 of set (2,3) which results in the path (4,2,3,5) with $dp_2 = 12$, $dp_3 = 10$ and $l(P) = 36$; vertex 4 is then moved to G_3 (the entry '12' is moved to subcolumn 2) which results in the path (1,2,3,4) with $dp_2 = 10$, $dp_3 = 12$



$$D = \begin{bmatrix} 0 & 10 & 24 & 20 & 34 \\ 10 & 0 & 14 & 12 & 24 \\ 24 & 14 & 0 & 12 & 10 \\ 20 & 12 & 12 & 0 & 20 \\ 34 & 24 & 10 & 20 & 0 \end{bmatrix}$$

V^E	(1,2)	(1,4)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)
1	0: -	0: -	10: -	10: -	-:20	24: -	20: -
2	-: 0	10: -	0: -	0: -	-:12^b	14: -	^{b}12: -
3	-:14	-:12^b	-: 0	-:12^b	0: -	0: -	-: 0
4	-:12^b	-: 0	12$\overset{A}{:}$12	-: 0	-: 0	12: -	0: -
5	-:24	-:20	-:10	-:20	10: -	-: 0	-: 0
dp_i, dp_j	0,24	10,20	12,10 10,12	10,20	10,20	24,0	20,10
l(i,j)	10	20	14	12	12	10	20
l(P)	34	50	36/36	42	42	34	50
B	36	no	-	no	no	36	no

Figure 9: Example 5.

and $l(P) = 36$. The entry '36' is made twice in row '$l(P)$', once for each path found in step 2.a.

The minimum path length (minimum entry in row '$l(P)$') found in step 4 is 34, which occurs for sets (1,2) and (3,4). Step 4 selects one of these sets, say set (1,2) and finds that it fails the distance criterion (15); i.e., $l(1,2) = 10 < |dp_1 - dp_2| = |0 - 24| = 24$. Step 5.b. notes that set (1,2) is marked as having a possible B condition and finds that $dk_2 = d2_2$ occurs for vertex 3 with an entry of 14, but the path (3,2,5) is recursive (paths (3,2) and (2,5) have the edge (2,3) in common), thus vertex 3 cannot be the q2 vertex even though it does satisfy (17). The third maximum entry in subcolumn 2 occurs for vertex 4 with $dk_3 = d2_3 = 12$. The path (5,2,4) is non-recursive; therefore vertex 4 is the q_2 vertex. Furthermore, inequality (17) still holds, so a B condition does exist.

Step 5.b.1) then finds the local radius to be $\frac{1}{2}(dp_2 + dq_2) = \frac{1}{2}(24 + 12) = 18$. The center of the path (5,2,4) is found to occur at point y on the graph in Figure 9, located 6 units from vertex 3 on edge (3,2). The value of $dp_2 + dq_2 = 36$ is entered in the 'B' row of column (1,2) and the algorithm returns to step 4 to select the next minimum path length.

The next minimum path length is also 34 and occurs for set (3,5) as previously noted. Conditions very similar to those just discussed for set (1,2) are found to exist in set (3,5), with vertex 4 being the q vertex again. In this case, however, the B condition path is found to be the path (4,3,1)

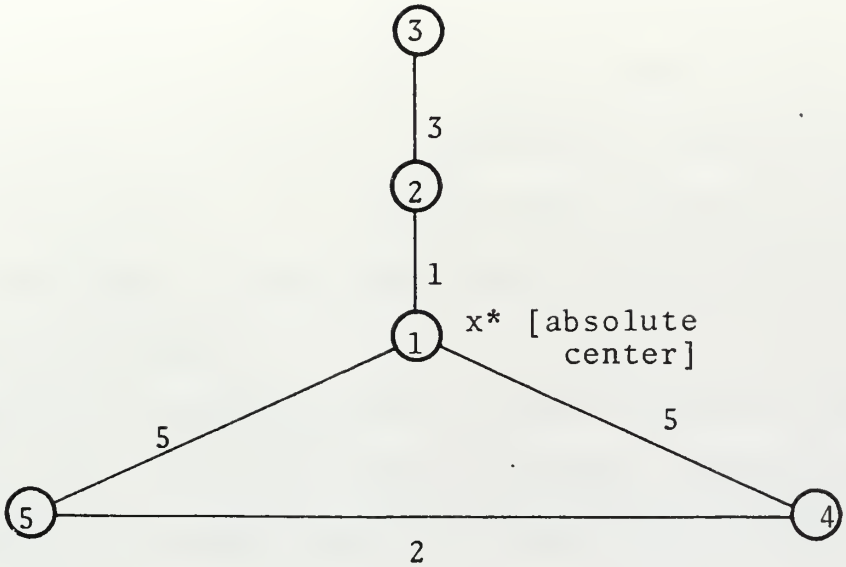
with midpoint at the point x on edge (2,3) of the graph; and the radius is again found to be 18.

Returning to step 4, the next minimum value of $l(P)$ = 36 is found to occur for both partitions under set (2,3). The two paths defined are (4,2,3,5) and (1,2,3,4), as previously noted; but these are the same paths defined by the B conditions in sets (1,2) and (3,5) respectively, so set (2,3) may be dropped from consideration. If they were not dropped they would be found to satisfy (15), and thus they give the two alternate central paths and absolute centers directly without any consideration of B conditions.

The next sets eligible for consideration are (2,4) and (3,4) with $l(P)$ = 42 for each, but this is strictly greater than the lengths of the B condition paths previously resolved for sets (1,2) and (3,5), so the algorithm terminates, having found the alternate central paths (4,2,3,5) and (1,2,3,4) with midpoints y and x respectively. The absolute radius is then $r(x) = r(y) = r_a = 18$.

Example 6, Figure 10, illustrates a case in which the central path cannot be defined by (14a) due to the existence of a circuit, as previously discussed. This case is characterized by a B condition in the minimum-path set as well as by the set's failure to satisfy (15).

The first iteration chooses either set (1,2) or (2,3) as the minimum-path set. If set (2,3) is chosen first, it is found to <u>not</u> contain a valid B condition as the paths (2,1,4) and (2,1,5) have the edge (1,2) in common. The set



$$D = \begin{bmatrix} 0 & 1 & 4 & 5 & 5 \\ 1 & 0 & 3 & 6 & 6 \\ 4 & 3 & 0 & 9 & 9 \\ 5 & 6 & 9 & 0 & 2 \\ 5 & 6 & 9 & 2 & 0 \end{bmatrix}$$

V \ E	(1,2)	(1,4)	(1,5)	(2,3)	(4,5)
1	0 : -	0 : -	0 : -	1 : -	5 $\overset{A}{:}$ 5
2	- : 0	1 : 0	1 : 0	0 : -	6 $\overset{A}{:}$ 6
3	- : 3	4 : -	4 : -	- : 0	9 $\overset{A}{:}$ 9
4	$_b5$: -	- : 0	- : 2	6 : -	0 : -
5	5 : -	- : 2	- : 0	$^b6$: -	- : 0
$\bar{dp}_i, dp_j$	5 , 3	4 , 2	4 , 2	6 , 0	9 , 0 0 , 9
l(i,j)	1	5	5	3	2
l(P)	9	11	11	9	11/11
B	10	-	-	no	-

Figure 10: Example 6.

is therefore dropped from consideration and set (1,2) is chosen on the second iteration.

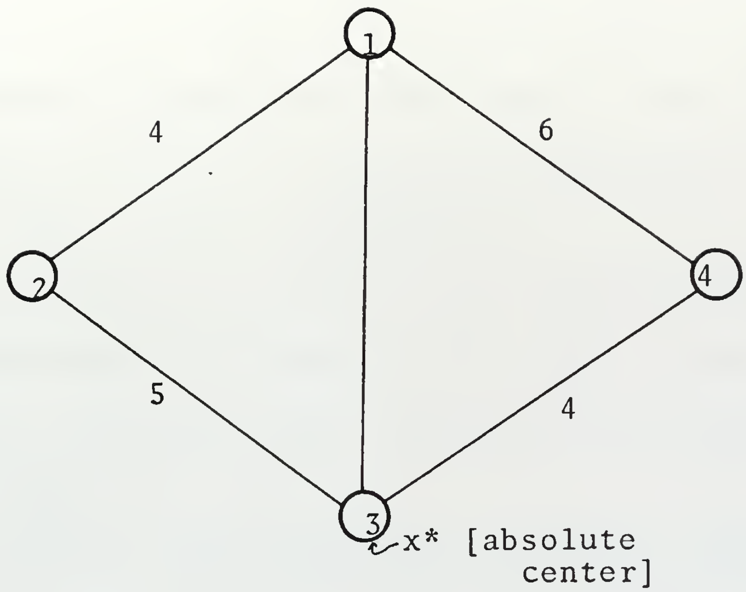
Set (1,2) defines the same initial paths as does set (2,3), but now the possible B condition is found to yield the valid B condition path (4,1,5) with a length of 10 units. The center of this path lies at vertex 1 and has a radius of five units.

The third iteration finds in step 4 that all remaining sets have $l(P) > 10$, and so the algorithm terminates in step 8 with an absolute radius of 5 and the absolute center at vertex 1.

Note in Figure 10 that the set (4,5) contains three type A ties, associated with Vertices 1, 2, and 3. When step 2.b. is called on to resolve these, the paths (3,2,1,4,5) and (3,2,1,5,4) are found. These two paths occur when all type A tied values are in the same subcolumn, and are the same paths defined by sets (1,4) and (1,5) respectively.

Example 7, Figure 11, illustrates a less obvious case generating both a type A tie and a B condition. Set (1,2) has a type A tie corresponding to vertex 3 and a B condition (which is never fully resolved) for the partition with vertex 3 in V_1 (subcolumn 1) of set (1,2). The first iteration through step 4 chooses set (3,4) which is found to have a B condition involving vertices 1 and 2 which yields the central path candidate (1,3,2) with a path length of 10. This gives a local center at vertex 3 with a corresponding radius of 5.

The second iteration through step 4 finds the partition in set (1,2) with $l(P) = 10$, but it is marked as having a



$$D = \begin{bmatrix} 0 & 4 & 5 & 6 \\ 4 & 0 & 5 & 9 \\ 4 & 5 & 0 & 4 \\ 6 & 9 & 4 & 0 \end{bmatrix}$$

V E	(1,2)	(1,3)	(1,4)	(2,3)	(3,4)
1	0 : -	0 : -	0 : -	4 : -	b5 : -
2	- : 0	4 : -	4 : -	0 : -	5 : -
3	b5 A 5	- : 0	- : 4	- : 0	0 : -
4	6 : -	- : 4	- : 0	- : 4	- : 0
dp_i, dp_j	b6 , 0 6 , 5	4 , 4	4 , 4	4 , 4	5 , 0
l(i,j)	4	5	6	5	4
l(P)	10/15	13	14	13	9
B	11/-	-	-	-	10

Figure 11: Example 7.

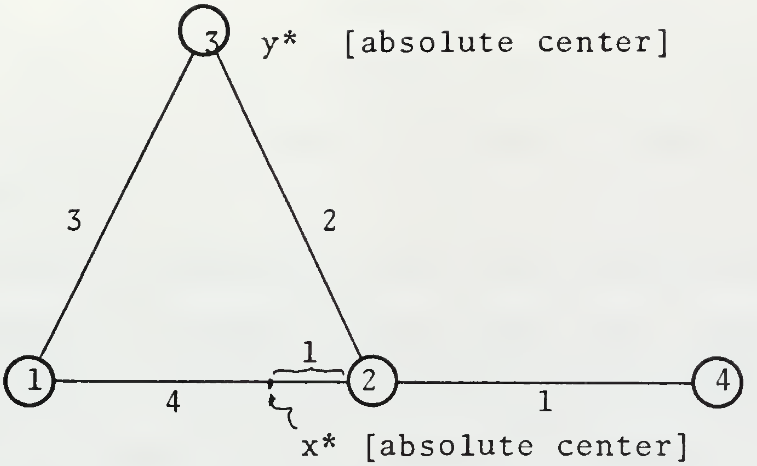
possible B condition. When this is resolved the B condition path (4,1,3) is found with a path length of 11, but this is strictly greater than the length of the central path candidate found in the previous iteration, so the set is dropped from further consideration.

The next iteration through step 4 shows that all remaining sets have l(P) values greater than the path found in the first iteration (10); therefore, the algorithm terminates in step 8.

The Rosenthal-Smith algorithm encounters trouble in dealing with peripheral vertices, where a peripheral vertex is defined as any vertex which is connected to the remainder of the graph via only one other vertex of the graph. In the calculation of the central path (v_x, v_y, v_z), if v_y is a peripheral vertex, all central path candidates associated with it have backpaths. The following three examples will illustrate how the arc-center approach eliminates this problem.

In Example 8, Figure 12, vertex 4 is a peripheral vertex and is connected to the rest of the graph by edge (2,4) via vertex 2. The arc-center computation tableau entries are all zero in subcolumn 2 of set (2,4) as they must be for any peripheral vertex.

The first iteration of the algorithm through step 4 chooses set (2,4) with a path length of 5 and resolves the possible B condition to find the B condition path (1,2,3) with a length of 6 units. This gives the local center x*



$$D = \begin{bmatrix} 0 & 4 & 3 & 5 \\ 4 & 0 & 2 & 1 \\ 3 & 2 & 0 & 3 \\ 5 & 1 & 3 & 0 \end{bmatrix}$$

V \ E	(1,2)	(1,3)	(2,3)	(2,4)
1	0 : -	0 : -	- : 3	4 : -
2	- : 0	- : 2^b	0 : -	0 : -
3	- : 2	- : 0	- : 0	$^b2$: -
4	- : 1^b	- : 3	1 : -	- : 0
dp_i, dp_j	0 , 2	0 , 3	1 , 3	4 , 0
$l(i,j)$	4	3	2	1
$\bar{l}(P)$	6	6	6	5
B	no	no	-	6

Figure 12: Example 8.

on edge (1,2) of the graph in Figure 12, with a local radius of 3 units.

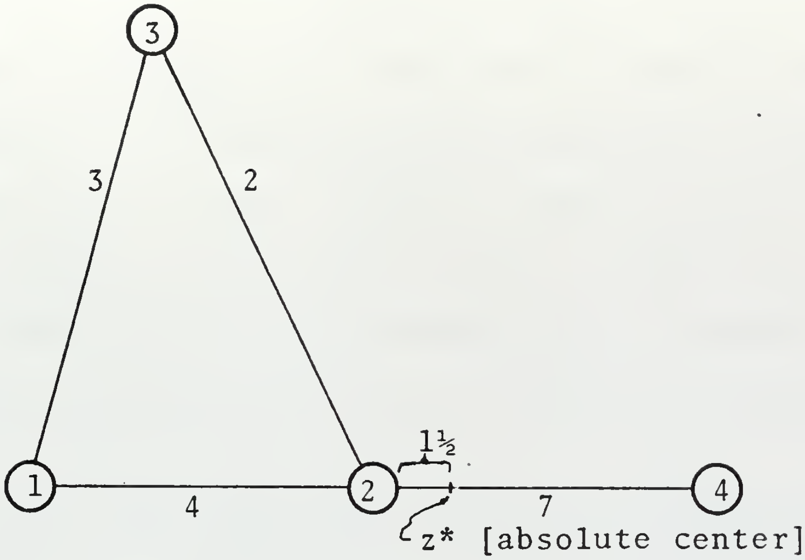
The three remaining sets all have l(P) = 6, so all must be resolved. Set (1,2) is found to not contain a valid B condition, and so defines the path (3,2,1). The set satisfies (15), so it is retained and found to again yield the local center x*.

The algorithm finds that set (1,3) does not contain a valid B condition but does satisfy (15) also. In this case the set is found to define the path (4,2,3,1) with midpoint y* located at vertex 3 and a radius of 3 units.

Set (2,3) is found to satisfy (15) on the last iteration. The path defined is (4,2,3,1), the same as that defined by set (1,3). Therefore, the absolute radius of the graph is 3, with alternate absolute centers, points x* and y*.

In this example, 3 arc-centers were found. The first, edge (1,2), was found through a B condition path first, and then through its own set, which satisfied (15). Since the alternate absolute center y* occurs at a vertex, two arc-centers, edges (1,3) and (3,2), were found for it.

Example 9, Figure 13, is a variation of Example 8 with l(2,4) increased to seven units. In this case sets (1,2) and (2,4) tie for minimum path length with l(P) = 11, but set (1,2) does not satisfy (15) while set (2,4) does. Both sets define the path (1,2,4) which has its midpoint at point z*. The remaining sets both have l(P) > 11, and so the



$$D = \begin{bmatrix} 0 & 4 & 3 & 11 \\ 4 & 0 & 2 & 7 \\ 3 & 2 & 0 & 9 \\ 11 & 7 & 9 & 0 \end{bmatrix}$$

V \ E	(1,2)	(1,3)	(2,3)	(2,4)
1	0 : -	0 : -	- : 3	4 : -
2	- : 0	- : 2^b	0 : -	0 : -
3	- : 2^b	- : 0	- : 0	$^b2$: -
4	- : 7	- : 9	7 : -	- : 0
dp_i, dp_j	0 , 7	0 , 9	7 , 3	4 , 0
l(i,j)	4	3	2	7
$\bar{l}$(P)	11	12	12	11
B	no	no	-	no

Figure 13: Example 9.

algorithm terminates with z* as the absolute center and $r_a = 5\frac{1}{2}$. There were no B condition sets.

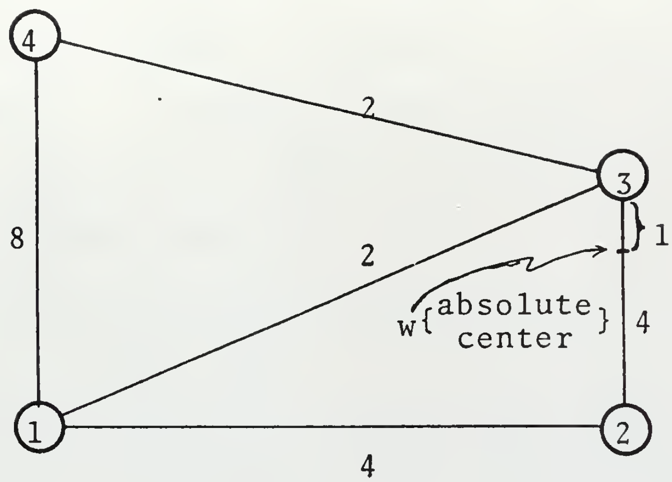
Example 10, Figure 14, illustrates a case in which a vertex (vertex 4) is technically not peripheral because it is connected to two other vertices, 1 and 3, in the graph. However, l(1,4) is greater than l(1,3) + l(3,4) (8 > 2 + 2 = 4), therefore all minimum distance paths to vertex 4 must pass through vertex 3, and vertex 4 behaves as a peripheral vertex.

There are three sets which tie with a minimum path length of six. These are: (1,3) and (3,4) which fail to satisfy (15) but do not contain B conditions; and (2,3), which does satisfy (15). Set (2,3) defines two alternate paths, (2,3,1) and (2,3,4), both having the same path length; but since (2,3) satisfies (15), the absolute center is on edge (2,3) and it is immaterial which path is considered. The absolute center is therefore a point w and the absolute radius is 3.

Examples 9 and 10 illustrated that no further special consideration need by given to peripheral vertices. This is a definite improvement over the Rosenthal-Smith algorithm, which, as previously noted, requires special inputs for peripheral vertices when programmed into a computer.

D. EXTENSION TO THE ABSOLUTE M-CENTER

Define the arc-m-center of the graph G(V,A) to be a set of m edges A_m^* such that for every other set of m edges $A_m \varepsilon G$,



$$D = \begin{bmatrix} 0 & 4 & 2 & 4 \\ 4 & 0 & 4 & 6 \\ 2 & 4 & 0 & 2 \\ 4 & 6 & 2 & 0 \end{bmatrix}$$

V \ E	(1,2)	(1,3)	(1,4)	(2,3)	(3,4)
1	0 : -	0 : -	0 : -	- : 2	$^b2$: -
2	- : 0	4 $\overset{A}{.}$ -	4 : -	0 : -	4 : -
3	$^b2$: -	- : 0	$^b2$ $\overset{A}{.}$ 2	- : 0	0 : -
4	4 : -	- : 2^b	- : 0	- : 2^b	- : 0
dp_i, dp_j	4 , 0	4 , 2 0 , 4^b	$^b4$, 2 4 , 0	0 , 2	4 , 0
1(i,j)	4	2	8	4	2
1(P)	8	8 / 6	14/12	6	6
B	no	- /no	- /no	no	no

Figure 14: Example 10.

$$\max_{v \varepsilon V} d(v,A_m) \geq \max_{v \varepsilon V} d(v,A_m^*) = r_m'. \tag{18}$$

Where $d(v,A_m)$ is the minimum distance from vertex v to any endpoint (defined vertex) of the set A_m. Since every edge has two endpoints, A_m defines a set of m pairs of adjacent vertices V_{2m}. Inequality (18) can therefore be written in a form consistent with (5):

$$\max_{v \varepsilon V} d(v,V_{2m}) \geq \max_{v \varepsilon V} d(v,V_{2m}^*) = r_m'. \tag{19}$$

$l(A_m^*)$ may now be defined by generalizing (14a) in the following form (using the same notation and definitions):

$$l(A_m^*) = \min_{A_m \varepsilon A} \left\{ \max_{(i,j) \varepsilon A_m} [dp_i + dp_j + l(i,j)] \right\}. \tag{20}$$

Equation (20) outlines an iterative algorithm for partioning G into a set of m subgraphs $\{G_i\}$, i = 1, ... ,m, and simultaneously finding the central path and hence the absolute radius $r(G_i)$, of each G_i. As the algorithm iterates through all possible partitionings of G (A_m A), that partition yielding the minimum absolute radius is found. Since the set of points X_m^* yielding the minimum absolute radius defines an optimum partition of G $\{G_i^*\}$, and since the algorithm will test all possible partitions of G, the algorithm must find $\{G_i^*\}$.

The computational procedure is essentially the same as previously outlined for the single arc-center algorithm, and iterates through $\binom{M}{m}$ modified arc-center calculations. The optimum result from the set of all results is then chosen.

The changes required in the arc-center algorithm are listed below.

1. In step 1, the algorithm now starts by choosing one of the $\binom{M}{m}$ combinations of pairs of adjacent vertices not previously examined; these are the sets V_{2m}.

a. Since there are $k \leq 2m$ different vertices involved, the computation tableau may be reduced by k rows. This is possible because all entries in rows corresponding to elements of the set under consideration have zero entries.

b. The minimum distance from each vertex (row in D) to any element of V_{2m} is entered in <u>all</u> subcolumns corresponding to that element. This produces ties any time two or more edges sharing a common vertex are in A_m. Call these a type C tie.

2. No Change.

a. No Change; but there could be a type A multiple tie; i.e., one vertex could be equidistant between several elements of V_{2m}. It is still allowed in only one set at a time, and checked in the same manner as in the arc-center algorithm.

b. No Change; same comment.

3. Compute l(P) for each subset with and without type C ties. For each group of sets with a common type C, choose one with, rest without to min(max l(P)).

4. Examine all tying sets generated by procedure A in step 2 and choose those with minimum path length for further consideration. Do <u>not</u> discard the others at this time.

5. a. No Change.

b. If no condition B exists, retain the subset as originally defined, and process it as though it had satisfied

(15) except the center of the path is simply the midpoint of (p_i, i, j, p_j).

1). No Change.

2). Delete.

6. No Change.

7. No Change.

8. If all m subsets have not been processed, return to step 4.

9. If any type C ties remain, discard the entire set and return to step 1.

10. Find the maximum radius in the set of m radii.

a. If this radius is less than the maximum radius retained from any previous iteration, discard all previous results and store all results of this iteration.

b. If this radius equals that retained from any previous iterations, retain the previous results and store the present results also.

c. If this radius is greater than that retained from any previous iterations, discard the results from this iteration.

11. If all $\binom{M}{m}$ combinations have not been examined, return to step 1.

The analyst now may choose between any sets of m-centers which may have tied for min(max radius), using some further criterion.

E. EXAMPLE

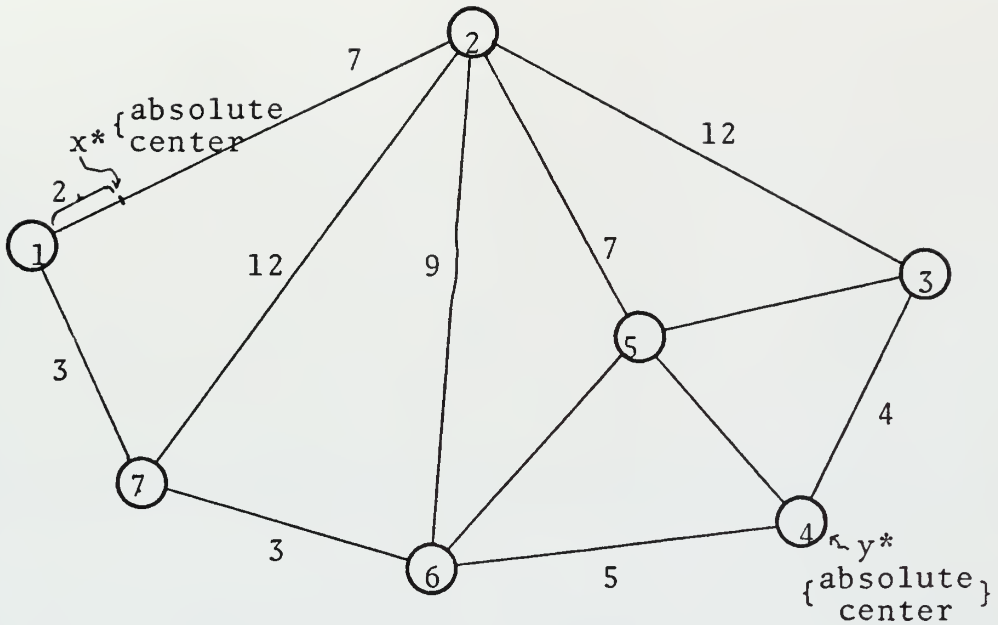
Example 11, Figure 15a, is an abbreviated absolute 2-center problem which will illustrate the application of the

arc-m-center algorithm. Only six of the possible $\binom{12}{2} = 66$ sets $V_{2 \cdot 2}$ are shown in Figure 15b. Note that in each set (iteration), the pair identity of the vertices is retained; they cannot be mixed randomly. Since three sets are shown side by side in each tableau, it was not possible to delete as many rows from the tableaux as would normally be done if a separate tableau were made for each set.

Observe that in set 1, the set {(1,2), (4,6)}, vertex 7 is equidistant between vertices 1 and 6. While this is not a type C tie as previously defined since two different vertices are involved, it is handled in the same manner. In this instance subset (4,6) is not affected when vertex 7 is removed; and since its path length is greater than that of subset (1,2) with vertex 7 included, it makes no difference which subset includes vertex 7.

Set 5 is entirely different. Here vertex 4 is equidistant between vertices 5 and 6, producing both type A and C ties. When vertex 4 is in subset (6,7), the path (4,6,7,1) is defined with a length of 11; and (5,6) defines the type B path (2,5,3), with a length of 13. When vertex 4 is moved to subset (5,6), a type A tie is produced which is resolved into path (4,6,5,2) with length 16, which satisfies (15); or the above mentioned type B path (2,5,3), with vertex 4 being "peripheral" to the path.

Set 4 is the set which satisfies min(max radius) of those sets shown in Figure 15b and may be shown to be the optimum set. Subset (1,2) defines $G_1 = \{1,2,7\}$ with absolute center at point x; and (3,4) defines $G_2 = \{3,4,5,6\}$ with



$$
D = \begin{bmatrix}
0 & 7 & 15 & 11 & 10 & 6 & 3 \\
7 & 0 & 12 & 12 & 7 & 9 & 10 \\
15 & 12 & 0 & 4 & 6 & 9 & 12 \\
11 & 12 & 4 & 0 & 5 & 5 & 8 \\
10 & 7 & 6 & 5 & 0 & 4 & 7 \\
6 & 9 & 9 & 5 & 4 & 0 & 3 \\
3 & 10 & 12 & 8 & 7 & 3 & 0
\end{bmatrix}
$$

Figure 15a: Example 11.

V^E	SET I (1,2)	(4,6)	SET II (1,2)	(4,5)	SET III (2,7)	(4,5)
1	- : -	- : -	- : -	- : -	- : 3	- : -
3	- : -	4 : -	- : -	4 : -	- : -	4 : -
5	- : -	- : 4	- : -	- : -	- : -	- : -
6	- : -	- : -	- : -	- : 4	- : 3	- : -
7	3 : - C	- : 3	3 : -	- : -	- : -	- : -
dp_i, dp_j	3 , 0	4 , 4	3 , 0	4 , 4	0 , 3	4 , 0
l(i,j)	7	5	7	5	12	5
l(P)	10	13	10	13	15	9
B	-	-	-	-	-	-
w/o C	7	13	-	-	-	-

V^E	SET IV (1,2)	(3,4)	SET V (5,6)	(6,7)	SET VI (2,7)	(2,5)
1	- : -	- : -	- : -	- : 3	- : 3	- : -
2	- : -	- : -	7 : -	- : -	- : -	- : -
3	- : -	- : -	6 : -	- : -	- : -	- : 6
4	- : -	- : -	5 $\overset{A}{:}$ 5	C 5 : -	- : -	- : 5
5	- : -	- : 5	- : -	- : -	- : -	- : -
6	- : -	- : 5^b	- : -	- : -	- : 3	- : -
7	3 : -	- : -	- : -	- : -	- : -	- : -
dp_i, dp_j	3 , 0	0 , 5	^{b}7 , 5 7 , 0	5 , 3	0 , 3	0 , 6
l(i,j)	7	4	4	3	12	7
l(P)	10	9	16/11	11	15	13
B	-	10	-/13	-	-	-
w/o C	-	-	-/13	6	-	-

Figure 15b: Example 11

absolute center at point y*, colocated with vertex 4. Both have an absolute radius of 5.

VI. CONCLUSION

A. SUMMARY

A brief review of known previous algorithms for the solution of the absolute m-center problem has been given, and mention made of the problems encountered in each. In addition, two new algorithms were presented. These algorithms are applicable to the solution of optimal location problems which can be structured in a graph theoretic manner, and so posed as absolute center problems.

The first of the algorithms, the vertex-m-center approach, was presented only as an aid in understanding the problem and for use in hand calculation of solutions to simple examples, particularly with $m = 1$. It is probably less efficient than the Rosenthal-Smith algorithm, but it does provide a basis for understanding and judging other algorithms.

The second algorithm presented was the arc-m-center algorithm. It was originally formulated as a means of bypassing the path check requirements of the Rosenthal-Smith algorithm. As the reader has seen, the effort was unsuccessful to the point that, for $m = 1$, the algorithm is quite possibly significantly less efficient than the Rosenthal-Smith algorithm. However, for $m > 1$, it is felt that the arc-m-center approach may prove more efficient and more easily implemented than any previous algorithm. This algorithm can be made more efficient, both in terms of data

management and of required operational steps, than outlined herein (see following section); and a final choice between it and the Rosenthal-Smith algorithm, or a meld of the two, would necessarily be based on operational results.

B. AREAS FOR FURTHER CONSIDERATION

As previously noted, the vertex-m-center approach was extended to cases with $m > 1$ by an unproven method. Further research is needed to determine under what conditions the use of the vertex-m-center to partition a graph may yield an optimum partition, and a more efficient means of selecting the optimum partition when the vertex m-center is not unique.

This method does provide an upper bound on the solution which might be useful in conjunction with some other algorithms to eliminate some of the possible solutions as being infeasible, and thus shorten the number of computations and time required. It might also provide a more efficient starting point for the Rosenthal-Smith algorithm than the m-node divisional path presently used. There is certainly a need for more investigation into these areas.

Another approach to solving the absolute m-center problem which might be worthy of investigation is to consider each of the $\binom{n}{m}$ partitions of G about the central vertices of the m central paths, using the vertex m-center radius as an upper bound to reduce the number of iterations which are carried to completion.

In all previous work on the absolute m-center problem except that of Rosenthal and Smith (see comment, Section III.G), the only criterion has been to minimize the maximum value of the set of m radii. This frequently yields a number of alternate solutions, all having the same maximum radius with different lesser radii. It is felt that attention should be given to further criteria for choosing <u>the</u> optimum solution. Two possible considerations are to minimize the sum of the radii or to minimize the absolute differences between radii.

LIST OF REFERENCES

1. Dreyfus, S. E., "An Appraisal of Some Shortest Route Algorithms," Operations Research, v. 17, p. 395-412, May-June 1969.

2. Francis, R. L., "Some Aspects of a Minimax Location Problem," Operations Research, v. 12, p. 1163-1168, November 1967.

3. Gillespie, C. M. Jr., Locating Absolute 2-Centers of Undirected Graphs, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1968.

4. Hakimi, S. L., "Optimum Locations of Switching Centers and the Absolute Center and Medians of a Graph," Operations Research, v. 12, p. 450-459, July-August 1964.

5. Hakimi, S. L., "Optimum Distribution of Switching Centers and Some Graph Theoretic Problems," Operations Research, v. 13, p. 462-475, July-August 1965.

6. Ore, O., Theory of Graphs, American Mathematical Society, Providence, Rhode Island, 1962.

7. Rosenthal, M. R., and Smith, S. B., The M-Center Problem Paper No. TP2.1 presented at 31st National ORSA Meeting, New York, New York, May-June 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst Professor A. W. McMasters, Code 55 [Mg] Department of Operations Analysis Naval Postgraduate School Monterey, California 93940	5
4.	Department of Operations Analysis, Code 55 Naval Postgraduate School Monterey, California 93940	1
5.	LT John J. Reed, USN 323 Bench Boulevard Billings, Montana 59100	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Two Algorithms for Finding the Absolute M-Center of a Graph

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Master's Thesis; March 1971

5. AUTHOR(S) *(First name, middle initial, last name)*

John J. Reed

6. REPORT DATE March 1971	7a. TOTAL NO. OF PAGES 75	7b. NO. OF REFS 7
8a. CONTRACT OR GRANT NO b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Two algorithms for finding the absolute m-center are developed, combining the ideas of Hakimi, Gillespie, and Rosenthal and Smith. The first algorithm developed is essentially a hand-computational method. It is based on partitioning the graph into m subgraphs centered on the elements of the vertex m-center. The minimum distance tree rooted on each element of the vertex m center is then formed and modified to yield the central path and thus the absolute center of each subgraph. This algorithm will give the absolute m-centers of a graph if each of these m-central paths passes through an element of the vertex m-center. The second algorithm is an iterative search of all possible sets of m edges on which the absolute m-center may be located. It is less efficient than the algorithm of Rosenthal and Smith when m = 1, but appears to be more efficient for m > 1. It does eliminate the problems encountered by the Rosenthal-Smith algorithm in handling peripheral vertices.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

UNCLASSIFIED
Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Multi-Center						
M-Center						
Vertex Center						
Absolute Center						
Location Problem						
Graph Theory						
Network Theory						
Central Path						

Thesis
R2695
c.1

126279

Reed

Two algorithms for
finding the absolute
m-center of a graph.